Содержание

Must have 2

Задача A. Умножение многочленов [0.4 sec, 256 mb] 2

Задача B. Деление многочленов [0.4 sec, 256 mb] 3

Задача C. Умножение чисел [0.9 sec, 256 mb] 4

Обязательные задачи 5

Задача D. Раздвоение [0.7 sec, 256 mb] 5

Задача E. Дуэль [1 sec, 256 mb] 6

Задача F. 4-SUM [2.5 sec, 256 mb] 7

Задача G. Длинный наибольший общий делитель [5 sec, 256 mb] 8

Дополнительные задачи 9

Задача H. Умножение четырех чисел [1 sec, 256 mb] 9

Задача I. Уравнение [6 sec, 256 mb] 10

Задача J. АВЛ-деревья [0.3 sec, 256 mb] 11

Задача K. ДНК роботов [1 sec, 256 mb] 12

Обратите внимание, входные данные лежат в **стандартном потоке ввода** (он же stdin), вывести ответ нужно в **стандартный поток вывода** (он же stdout).

В некоторых задачах большой ввод и вывод. Пользуйтесь быстрым вводом-выводом.

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

Обратите внимание на `GNU C++` компиляторы с суффиксом `inc`, они позволяют пользоваться дополнительной библиотекой. Под ними можно сдать вот это.

Must have

Задача A. Умножение многочленов [0.4 sec, 256 mb]

Даны многочлены $P(x)$ и $Q(x)$, найдите $P(x)Q(x)$. $Q(x) = x^k - 1$.
У $P(x)$ все коэффициенты — случайные целые числа от 0 до $m - 1$.
Все вычисления происходят по модулю m, m — простое от 2 до $10^9 + 7$.

Формат входных данных

Первая строка содержит целые числа n, k, m.
Следующая строка содержит n чисел $p_0, p_1, \ldots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Формат выходных данных

Выведите $n + k$ целых чисел от 0 до $m - 1$: $b_0, b_1, \ldots, b_{n+k-1}$,
такие, что $P(x)Q(x) = \sum b_i x^i \mod m$.

Система оценки

$1 \leqslant n, k \leqslant 10^5$.

Примеры

stdin	stdout
4 1 7 1 2 3 4	6 6 6 6 4

Задача B. Деление многочленов [0.4 sec, 256 mb]

Даны многочлены $P(x)$ и $Q(x)$, найдите такие $A(x)$ и $R(x)$, что $P(x) = A(x)Q(x) + R(x)$ и $degR < degQ$. $Q(x) = x^k - 1$. У $P(x)$ все коэффициенты — случайные целые числа от 0 до $m - 1$. Все вычисления происходят по модулю m, m — простое от 2 до $10^9 + 7$.

Формат входных данных

Первая строка содержит целые числа n, k, m.
Следующая строка содержит n чисел $p_0, p_1, \ldots, p_{n-1}$, $P(x) = \sum p_i x^i$.

Формат выходных данных

На первой строке выведите $t = \max(n-k, 1)$ целых чисел от 0 до $m-1$: $a_0, a_1, \ldots, a_{t-1}$. На второй строке k целых чисел от 0 до $m-1$: $r_0, r_1, \ldots, r_{k-1}$. Все эти числа должны обладать свойством, что $P(x) = Q(x)(\sum a_i x^i) + \sum r_i x^i \mod m$.

Система оценки

$1 \leqslant n, k \leqslant 10^5$.

Примеры

stdin	stdout
3 1 7 1 5 1	6 1 0
3 1 7 2 4 1	5 1 0
3 1 7 0 4 1	5 1 5

Задача C. Умножение чисел [0.9 sec, 256 mb]

Требуется перемножить два целых неотрицательных числа.

Формат входных данных

В двух строках даны два целых неотрицательных числа в 10-чной системе счисления. Максимальная длина числа $= 2^{18}$.

Формат выходных данных

Выведите в выходной файл произведение.

Пример

stdin	stdout
13 100	1300

Замечание

В этой задаче зайдут и Фурье, и Карацуба.

Фурье вы сможете переиспользовать в следующих задачах.

Обязательные задачи

Задача D. Раздвоение [0.7 sec, 256 mb]

Обозначим две последовательности действительных чисел $x(k)$ и $y(k)$.
Определим последовательность комплексных чисел $z(k)$: $z(k) = x(k) + iy(k)$.

Пусть $FFT_N(k, z) = \sum_{n=0}^{N-1} z_n e^{2\pi ikn/N}$.

Аналогичным образом определяются $FFT_N(k, x)$ и $FFT_N(k, y)$.
По вычисленным значениям $FFT_N(k, z)$ восстановите значения $FFT_N(k, x)$ и $FFT_N(k, y)$.

Формат входных данных

В первой строке входного файла записано целое число N ($1 \leqslant N \leqslant 2^{30}$, N является степенью двойки). Далее следуют целые неотрицательные числа A, B, C, D, E, F, не превосходящие 1000. Для экономии времени ввода значения $FFT_N(k, z)$ нужно будет вычислять по следующим формулам:

$FFT_N(k, z).real = ((A + B \cdot k)\ xor\ (C \cdot k)) \cdot 10^{-3}$,
$FFT_N(k, z).imag = ((D + E \cdot k)\ xor\ (F \cdot k)) \cdot 10^{-3}$,
где $FFT_N(k, z).real$ и $FFT_N(k, z).imag$ — действительная и мнимая части соответственно.

Затем дано число M — количество запросов ($1 \leqslant M \leqslant 10^5$).
Далее следуют M целых чисел q_j ($0 \leqslant q_j < N$).

Формат выходных данных

В выходной файл выведите M строк. В j-ой строке — значения $FFT_N(q_j, x)$ и $FFT_N(q_j, y)$.
Значения должны отличаться от правильных не более, чем на 10^{-4}.

Примеры

stdin	stdout
2 1000 0 0 0 0 0 2 0 1	1.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
4 0 100 300 500 100 200 4 0 1 2 3	0.000 0.000 0.500 0.000 0.504 0.140 0.516 0.176 0.656 0.000 0.812 0.000 0.504 -0.140 0.516 -0.176
1048576 999 998 997 996 995 994 3 17 239239 2011	540.737 -1587.741 1589.778 539.689 2404.809 531.421 1359.578 1569.751 3678.277 -523.243 526.382 3664.887

Задача E. Дуэль [1 sec, 256 mb]

Двое дуэлянтов решили выбрать в качестве места проведения поединка тёмную аллею. Вдоль этой аллеи растёт n деревьев и кустов. Расстояние между соседними объектами равно одному метру. Дуэль решили проводить по следующим правилам. Некоторое дерево выбирается в качестве стартовой точки. Затем два дерева, находящихся на одинаковом расстоянии от исходного, отмечаются как места для стрельбы. Дуэлянты начинают движение от стартовой точки в противоположных направлениях. Когда соперники достигают отмеченных деревьев, они разворачиваются и начинают стрелять друг в друга.

Дана схема расположения деревьев вдоль аллеи. Требуется определить количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Формат входных данных

Во входном файле содержится одна строка, состоящая из символов '0' и '1' — схема аллеи. Деревья обозначаются символом '1', кусты — символом '0'. Длина строки не превосходит 100 000 символов.

Формат выходных данных

Выведите количество способов выбрать стартовую точку и места для стрельбы согласно правилам дуэли.

Примеры

stdin	stdout
101010101	4
101001	0

В первом примере возможны следующие конфигурации дуэли (стартовое дерево и деревья для стрельбы выделены жирным шрифтом): **1**0**1**0**1**0101, 10**1**0**1**0**1**01, 1010**1**0**1**0**1** и **1**010**1**0101.

Задача F. 4-SUM [2.5 sec, 256 mb]

Дан массив длины n из целых чисел число S. Выберите любые $1 \leqslant i, j, k, l \leqslant n$: $a_i + a_j + a_k + a_l = S$.

Формат входных данных

На первой строке n ($4 \leqslant n \leqslant 256\,000$) и S ($0 \leqslant S \leqslant 256\,000$).
В следующей строке массив из n число от 0 до 256 000.

Формат выходных данных

Четвёрку индексов. Одно число можно брать два раза.
Если таких i, j, k, l не существует, выведите 0.

Пример

stdin	stdout
5 21 0 10 11 13 12	1 2 1 3
4 0 1 1 1 1	0

Подсказка по решению

Чтобы не было переполнений, можно считать по случайному простому модулю.

Задача G. Длинный наибольший общий делитель [5 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

stdin	stdout
10 15 1000000000000000000 1152921504606846976 17 100	5 262144 1

Подсказка по решению

Пусть длина входного числа в двоичной системе счислении равна n. У вас есть бинарный алгоритм поиска `gcd`, который делает не более n^2 операций, Чтобы это решение зашло по времени, нужно выбрать систему счисления $B = 10^9$ и все базовые операции с числами реализовать за $\mathcal{O}$(длины числа в системе счисления B).

Дополнительные задачи

Задача H. Умножение четырех чисел [1 sec, 256 mb]

Требуется перемножить четыре целых положительных числа.

Формат входных данных

В четырех строках даны четыре целых положительных числа в десятичной системе счисления. Числа даны без ведущих нулей. Суммарная длина всех чисел не превосходит 2^{20} цифр.

Формат выходных данных

Выведите в выходной файл произведение данных чисел.

Пример

stdin	stdout
13 100 5 2	13000

Замечание

Решение жюри на C++ работает 0.204 секунды.

Задача I. Уравнение [6 sec, 256 mb]

Дано уравнение вида $X^N + Y^N \equiv Z^N \mod M$.

Требуется для фиксированных N и M найти количество различных решений этого уравнения. Решением назовём такую тройку натуральных чисел (X, Y, Z), что выполняется:

- $1 \leqslant X \leqslant Y < M$
- $1 \leqslant Z < M$
- $X^N + Y^N \equiv Z^N \mod M$

Формат входных данных

В единственной строке входного файла записаны числа N и M ($1 \leqslant N \leqslant 7^7, 1 \leqslant M \leqslant 7^7$).

Формат выходных данных

В выходной файл выведите одно число — ответ на задачу.

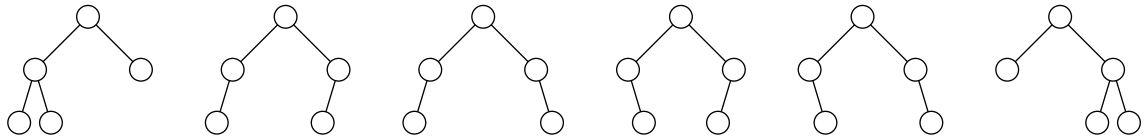
Примеры

stdin	stdout
1 3	2
2 4	5
3 5	8

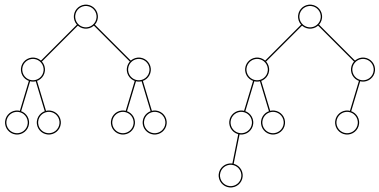
Задача J. АВЛ-деревья [0.3 sec, 256 mb]

АВЛ-дерево — сбалансированное по высоте двоичное дерево поиска: для каждой его вершины высота её двух поддеревьев различается не более чем на 1. АВЛ-деревья названы по первым буквам фамилий их изобретателей, Г. М. Адельсона-Вельского и Е. М. Ландиса.

Для фиксированного количества вершин может существовать несколько АВЛ-деревьев. Например, существует шесть АВЛ-деревьев, состоящих из пяти вершин.



Также деревья с одинаковым количеством вершин могут иметь различную высоту. Например, существуют деревья из семи вершин с высотами 2 и 3 соответственно.



Требуется по заданным n и h найти количество АВЛ-деревьев, состоящих из n вершин и имеющих высоту h. Так как ответ может быть очень большим, требуется найти остаток от деления искомого количества на 786433.

Формат входных данных

Во входном файле даны числа n и h ($1 \leqslant n \leqslant 65535$, $0 \leqslant h \leqslant 15$).

Формат выходных данных

Выведите одно число — остаток от деления количества АВЛ-деревьев, состоящих из n вершин и имеющих высоту h, на 786433.

Пример

stdin	stdout
7 3	16

Замечание

786433 — простое число, $786433 = 3 \cdot 2^{18} + 1$.

Задача K. ДНК роботов [1 sec, 256 mb]

Последние достижения в технологии синтеза ДНК позволили провести эксперимент по созданию биороботов.

Для облегчения задачи создания ПО для управления роботами было принято решение, что их ДНК будет состоять из $M = 2^n$ символов для некоторого $n \geqslant 2$. Кроме этого, по техническим причинам это будет не обычная строка, а циклическая, то есть её можно начинать читать с любой позиции.

Одной из целей эксперимента является изучение мутаций биороботов. В результате продолжительных наблюдений было найдено много различных видов роботов. Для понимания процесса мутации учёным необходимо решить следующую задачу. Для ДНК двух роботов требуется определить коэффициент их похожести. Он вычисляется, как максимальное количество совпадающих символов при наилучшем совмещении этих ДНК. Чем больше символов совпадает, тем лучше совмещение.

Требуется написать программу, которая найдёт наилучшее совмещение двух ДНК.

Формат входных данных

В первой строке входного файла записано одно число M ($4 \leqslant M \leqslant 131\,072$). В следующих двух строках записаны ДНК двух роботов. Обе ДНК — строки, состоящие ровно из M символов из множества {'A', 'C', 'G', 'T'}.

Формат выходных данных

В выходной файл выведите два числа — максимальное количество совпадающих символов и значение оптимального сдвига — неотрицательное количество символов второй ДНК, которые необходимо перенести из конца строки в её начало для достижения наилучшего совмещения.

Пример

stdin	stdout
16 ACGTACGTACGTACGT CGTACGTACGTACGTC	15 1

Подсказка по решению

Идея решения этой задачи разобрана на практике. Чтобы получить OK в контесте нужно, чтобы константа вашего FFT и количество вызовов были оптимальны.