

Задача А. Представление целого числа в 16-битном типе данных

Имя входного файла: `binshort.in`
Имя выходного файла: `binshort.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во входном файле задано целое число. Выведите в выходной файл представление этого числа в 16-битном двоичном дополнительном коде.

Формат входных данных

В первой строке входного файла задано одно целое число n в десятичной записи ($-2^{15} \leq n < 2^{15}$).

Формат выходных данных

Выведите в выходной файл две строки. В первой строке выведите двоичную запись дополнительного кода для числа n . Во второй строке выведите шестнадцатеричную запись. Отделяйте записи соседних байтов пробелами. Следуйте формату, указанному в примере.

Примеры

| <code>binshort.in</code> | <code>binshort.out</code> |
|--------------------------|----------------------------|
| 3 | 00000000 00000011 00 03 |
| -57 | 11111111 11000111 FF C7 |

Задача В. Представление целого числа

Имя входного файла: `binint.in`
Имя выходного файла: `binint.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во входном файле задано целое число. Выведите в выходной файл представление этого числа в 32-битном двоичном дополнительном коде.

Формат входных данных

В первой строке входного файла задано одно целое число n в десятичной записи ($-2^{31} \leq n < 2^{31}$).

Формат выходных данных

Выведите в выходной файл две строки. В первой строке выведите двоичную запись дополнительного кода для числа n . Во второй строке выведите шестнадцатеричную запись. Отделяйте записи соседних байтов пробелами. Следуйте формату, указанному в примере.

Примеры

| <code>binint.in</code> | <code>binint.out</code> |
|------------------------|----------------------------------------------------|
| 3 | 00000000 00000000 00000000 00000011 00 00 00 03 |
| -57 | 11111111 11111111 11111111 11000111 FF FF FF C7 |

Задача С. Представление вещественного числа в 32-битном типе с плавающей точкой

Имя входного файла: `binflt32.in`
Имя выходного файла: `binflt32.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во входном файле задано вещественное число. Выведите в выходной файл представление этого числа в 32-битном типе `float`. В этом типе 1 старший бит хранит знак числа, следующие 8 битов отведены под экспоненту, а в оставшихся 23 битах находится мантисса.

Формат входных данных

В первой строке входного файла заданы два целых числа p и q через пробел ($|p|, |q| \leq 10^4, q \neq 0$). Число, которое нужно представить в типе `float` — это $\frac{p}{q}$. Помните, что если результат деления нельзя сохранить точно, в результирующей переменной типа `float` должно оказаться наиболее близкое из тех чисел, которые в этом формате можно хранить.

Формат выходных данных

Выведите в выходной файл две строки. В первой строке выведите двоичную запись представления данного числа. Во второй строке выведите шестнадцатеричную запись. Отделяйте записи соседних байтов пробелами. Следуйте формату, указанному в примере.

Число 0 следует записывать с положительным (то есть нулевым) знаковым битом.

Примеры

| <code>binflt32.in</code> | <code>binflt32.out</code> |
|--------------------------|----------------------------------------------------|
| 9 2 | 01000000 10010000 00000000 00000000 40 90 00 00 |
| 5 -3 | 10111111 11010101 01010101 01010101 BF D5 55 55 |

Задача D. Представление вещественного числа

Имя входного файла: `binfloat.in`
Имя выходного файла: `binfloat.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Во входном файле задано вещественное число. Выведите в выходной файл представле-

ние этого числа в 64-битном типе `double`. В этом типе 1 старший бит хранит знак числа, следующие 11 битов отведены под экспоненту, а в оставшихся 52 битах находится мантисса.

Формат входных данных

В первой строке входного файла заданы два целых числа p и q через пробел ($|p|, |q| \leq 10^9, q \neq 0$). Число, которое нужно представить в типе `double` — это $\frac{p}{q}$. Помните, что если результат деления нельзя сохранить точно, в результирующей переменной типа `double` должно оказаться наиболее близкое из тех чисел, которые в этом формате можно хранить.

Формат выходных данных

Выведите в выходной файл две строки. В первой строке выведите двоичную запись представления данного числа. Во второй строке выведите шестнадцатеричную запись. Отделяйте записи соседних байтов пробелами. Следуйте формату, указанному в примере.

Число 0 следует записывать с положительным (то есть нулевым) знаковым битом.

Примеры

| |
|----------------------------------------------------------------------------------------------------|
| <code>binfloat.in</code> |
| 9 2 |
| <code>binfloat.out</code> |
| 01000000 00010010 00000000 00000000 00000000 00000000 00000000 00000000 40 12 00 00 00 00 00 00 |
| <code>binfloat.in</code> |
| 5 -3 |
| <code>binfloat.out</code> |
| 10111111 11111010 10101010 10101010 10101010 10101010 10101010 10101011 BF FA AA AA AA AA AB |

Задача Е. Лишнее число

Имя входного файла: `excess.in`
Имя выходного файла: `excess.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В штаб секретной службы поступило сообщение от одного из агентов. Поступившее сообщение в зашифрованном виде представляет собой последовательность чисел, и лишь специальная программа способна расшифровать его и получить связный текст.

Обычно программа-расшифровщик быстро и бесшумно выдаёт связистам расшифрованный текст, но в этот раз вместо текста от программы поступил сигнал тревоги, свидетельствующий о том, что при пересылке сообщение было взломано или просто повреждено.

Корректное зашифрованное сообщение — это последовательность из $4 \cdot k$ целых чисел, в котором k различных чисел присутствуют по 4 раза каждое; для расшифровки даже не важны значения этих чисел, а важен лишь их порядок.

Однако, изучив зашифрованное сообщение, связисты обнаружили, что в нём $4 \cdot k + 1$ число. При этом ровно одно число является «лишним», то есть при его удалении зашифрованное сообщение становится корректным сообщением из $4 \cdot k$ чисел (возможно, четыре из них равны удалённому числу).

Связисты решили, что на будущее им нужна программа, которая находит такое «лишнее» число автоматически. Помогите им написать такую программу.

Формат входных данных

В первой строке входного файла задано число $n = 4 \cdot k + 1$, где n и k целые, и $1 \leq k \leq 10\,000$. Во второй строке строках находятся числа a_1, a_2, \dots, a_n , разделённые пробелами — зашифрованное сообщение. Известно, что $0 \leq a_i \leq 1\,000\,000$.

Формат выходных данных

В первую строку выходного файла выведите «лишнее» число из набора A_i .

Примеры

| | |
|------------------------|-------------------------|
| <code>excess.in</code> | <code>excess.out</code> |
| 5 4 1 4 4 4 | 1 |
| 9 1 3 3 1 3 3 1 1 | 3 |

Задача F. Квадратура круга

Имя входного файла: `squaring.in`
Имя выходного файла: `squaring.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

— Извини, Теодор, но это ты очень странно рассуждаешь. Бессмыслица — искать решение, если оно и так есть. Речь идёт о том, как поступать с задачей, которая решения не имеет. Это глубоко принципиальный вопрос, который, как я вижу, тебе, прикладнику, к сожалению, не доступен.

Аркадий и Борис Стругацкие, «Понедельник начинается в субботу»

Квадратура круга — задача, заключающаяся в нахождении построения с помощью циркуля и линейки квадрата, равновеликого данному кругу (то есть имеющего ту же площадь, что и круг). Наряду с трисекцией угла и удвоением куба, эта задача является одной из самых известных неразрешимых задач на построение с помощью циркуля и линейки. Однако, задача о квадратуре круга становится разрешимой, если расширить средства построения, а также если искать не точное, а приближённое решение.

В этой задаче требуется по кругу, заданному координатами центра и радиусом, построить квадрат, площадь которого отличается от площади этого круга не более чем на 10^{-6} .

В качестве средства предлагается использовать компьютер и один из доступных языков программирования.

Напомним, что площадь квадрата со стороной a равна a^2 , а площадь круга радиуса r равна $\pi \cdot r^2$, где $\pi \approx 3.1415926535897932384626433832795\dots$ — это половина длины окружности единичного радиуса.

Формат входных данных

В единственной строке входного файла заданы через пробел три целых числа x , y и r ($|x|, |y| \leq 100$, $1 \leq r \leq 100$) — координаты центра круга и его радиус.

Формат выходных данных

Выведите в выходной файл четыре строки. Каждая строка должна содержать два числа через пробел — координаты одной из вершин квадрата. Найденный квадрат должен иметь стороны, параллельные осям координат, и площадь, равную площади данного круга, а его центр должен совпадать с центром круга. В первой строке выведите координаты левой нижней вершины квадрата, во второй — левой верхней, в третьей — правой верхней и в четвёртой — правой нижней вершины.

Выводите вещественные числа как можно более точно! Допускается экспоненциальная форма вывода. При проверке ответов **все** проверки на равенство — сравнение координат точек и площадей квадратов — производятся с точностью до 10^{-6} .

Пример

| squaring.in | squaring.out |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 2 3 5 | -2.431134627264 -1.431134627264 -2.431134627264 7.431134627264 6.431134627264 7.431134627264 6.431134627264 -1.431134627264 |

Задача G. Разность между значениями

Имя входного файла: valdiff.in
Имя выходного файла: valdiff.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В выражении

$$f(x) = ax + \frac{x}{b}$$

известны коэффициенты a и b . Найдите разность значений функции $f(x)$ в двух точках x_1 и x_2 при условии, что $|x_2 - x_1| = 1$.

Формат входных данных

В первой строке входного файла заданы через пробел четыре целых числа a , b , x_1 и x_2 ($1 \leq a, b, x_1, x_2 \leq 10^9$). Гарантируется, что числа x_1 и x_2 отличаются ровно на единицу.

Формат выходных данных

Выведите в выходной файл одно число — требуемую разность. Абсолютная погрешность должна составлять не более 10^{-5} .

Примеры

| valdiff.in | valdiff.out |
|------------|-------------|
| 1 2 3 4 | -1.5 |
| 2 1 4 3 | 3 |

Задача H. Сумма значений функции

Имя входного файла: valsum.in
Имя выходного файла: valsum.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Найдите сумму значений функции

$$f(x) = x + \frac{1}{x}$$

в нескольких целых точках.

Формат входных данных

В первой строке входного файла задано целое число n — количество точек ($1 \leq n \leq 50$). В следующей строке заданы n целых чисел x_1, x_2, \dots, x_n через пробел — точки, значения функции в которых нужно просуммировать ($0 < |x_i| \leq 10^9$).

Формат выходных данных

Выведите в выходной файл одно число — сумму значений функции $f(x)$ в заданных точках. Ответ считается правильным, если абсолютная или относительная погрешность не превышает 10^{-9} .

Примеры

| valsum.in | valsum.out |
|------------|-------------------|
| 3 1 2 3 | 7.833333333333333 |
| 2 1 -1 | 0 |