

Эта тренировка предназначена для изучения некоторых частей стандартной библиотеки языка C++. Поэтому её условия более строгие, чем в пробной тренировке.

Во-первых, задачи получится послать на проверку только на языке C++. Во-вторых, следует стремиться к тому, чтобы большая часть решения состояла из вызовов библиотечных функций. Для этого вводится следующее формальное ограничение: в программе не должно быть циклов (`for`, `while` или `do...while`), рекурсивных функций, а также любых других конструкций, эквивалентных перечисленным и не являющихся вызовом библиотечных функций. Решения, нарушающие это правило, не будут считаться правильными при итоговом подсчёте баллов.

Задача А. Путешествие по прямой

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

Мы находимся на целочисленной прямой в точке A . Мы хотим попасть в точку B как можно скорее и остаться там. За одни сутки мы можем преодолеть расстояние не более d . Где мы окажемся через сутки после выхода из точки A ?

Формат входных данных

В единственной строке заданы через пробел три целых числа A , B и d — координаты точек и расстояние, которое можно преодолеть за сутки ($|A|, |B| \leq 10^9$, $1 \leq d \leq 10^9$).

Формат выходных данных

Выведите одно число — координату точки, в которой мы окажемся через сутки после выхода из точки A .

Примеры

linear-trip.in	linear-trip.out
4 7 2	6
4 7 5	7

Задача В. Степени числа

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

Дано число n . Найдите все числа от 1 до 10^9 включительно, которые можно представить в виде n^k , где k — целое неотрицательное число, и выведите их в порядке возрастания.

Формат входных данных

В единственной строке задано целое число n ($1 \leq n \leq 10^9$).

Формат выходных данных

Выведите все числа от 1 до 10^9 включительно, которые можно представить в виде n^k , где k — целое неотрицательное число, в порядке возрастания, разделяя соседние числа пробелом.

Примеры

small-powers.in	small-powers.out
300	1 300 90000 27000000
33333	1 33333

Задача С. Позиционные системы счисления

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

Задано число n в пятеричной записи. Вычислите сумму цифр в семеричной записи этого числа и выведите эту сумму как троичное число.

Формат входных данных

В единственной строке задано целое число n в пятеричной записи без ведущих нулей ($1 \leq n \leq 10^9$).

Формат выходных данных

Выведите одно число — сумму цифр в семеричной записи числа n , представленную в троичной записи без ведущих нулей.

Примеры

positional-notations.in	positional-notations.out
1	1
123	22

Пояснения к примерам

В первом примере $1_5 = 1_7$, сумма цифр равна 1_3 .

Во втором примере $123_5 = 1 \cdot 25 + 2 \cdot 5 + 3 \cdot 1 = 25 + 10 + 3 = 38$, в семеричной записи $38 = 5 \cdot 7 + 3 \cdot 1 = 53_7$, сумма цифр равна $8 = 2 \cdot 3 + 2 \cdot 1 = 22_3$.

Задача D. Список кратных чисел

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

Рассмотрим все числа от l до r включительно, делящиеся на d . Выведите их в порядке возрастания, если гарантируется, что их количество не превосходит 100.

Формат входных данных

В единственной строке заданы через пробел три целых числа l , r и d ($1 \leq l \leq r \leq 10^9$, $1 \leq d \leq 10^9$).

Формат выходных данных

Выведите все числа от l до r включительно, делящиеся на d , в порядке возрастания, разделяя соседние числа пробелом. Гарантируется, что входные данные таковы, что в правильном ответе не более 100 чисел.

Примеры

list-divisible.in	list-divisible.out
5 10 1	5 6 7 8 9 10
3 5 2	4

Задача E. Самая частая буква

Имя входного файла: most-common-letter.in
Имя выходного файла: most-common-letter.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана строка. Найдите букву, которая встречается в ней как можно больше раз. Если таких букв несколько, можно найти любую из них.

Формат входных данных

В единственной строке ввода задана строка длины от 1 до 100 символов включительно, состоящая из маленьких букв английского алфавита.

Формат выходных данных

Выведите букву, которая встречается в заданной строке максимальное количество раз. Если букв, встречающихся максимальное количество раз, несколько, выведите любую из них.

Примеры

most-common-letter.in	most-common-letter.out
aba	a
baacb	b

Задача F. Фильтрация букв

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

Заданы две строки s и t . Сотрите из s все буквы, которые хотя бы раз встречаются в t , и запишите оставшиеся буквы подряд в том порядке, в котором они следовали в s .

Формат входных данных

В первой строке ввода задана строка s , а во второй — строка t . Обе строки имеют длину от 1 до 100 символов включительно и состоят из маленьких букв английского алфавита.

Формат выходных данных

Выведите строку, которая получится из s , если стереть все буквы, которые хотя бы раз встречаются в t , а оставшиеся буквы записать без пропусков с сохранением их порядка.

Примеры

filter-letters.in	filter-letters.out
aba bb	aa
cabbage face	bbg

Задача G. Частые буквы

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

Даны строка s и число k . Найдите все маленькие буквы английского алфавита, встречающиеся в строке s хотя бы k раз, и выведите их по одному разу в лексикографическом порядке.

Формат входных данных

В первой строке ввода задана строка s , а во второй — целое число k ($1 \leq k \leq 100$). Строка s состоит из маленьких букв английского алфавита и имеет длину от 1 до 100 символов включительно.

Формат выходных данных

Выведите все маленькие буквы английского алфавита, которые встречаются в строке s хотя бы k раз, в лексикографическом порядке, разделяя соседние буквы пробелом.

Примеры

frequent-letters.in	frequent-letters.out
aba 1	a b
aba 2	a

Задача H. Самое частое число

Имя входного файла: most-common-number.in
Имя выходного файла: most-common-number.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана последовательность чисел. Найдите число, которое встречается в ней как можно больше раз. Если таких чисел несколько, можно найти любое из них.

Формат входных данных

В первой строке задано целое число n — длина последовательности ($1 \leq n \leq 100$). Во второй строке заданы целые числа a_1, a_2, \dots, a_n — элементы последовательности ($1 \leq a_i \leq 10^9$). Соседние числа в последовательности разделены пробелом.

Формат выходных данных

Выведите число, которое встречается в заданной последовательности максимальное количество раз. Если чисел, встречающихся максимальное количество раз, несколько, выведите любое из них.

Примеры

most-common-number.in	most-common-number.out
3 1 2 1	1
5 2 1 1 3 2	2

Задача I. Индекс после сортировки

Имя входного файла: index-after-sort.in
Имя выходного файла: index-after-sort.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана последовательность чисел. Запомните её первый элемент s , расположите числа в порядке (нестрогого) возрастания, после чего найдите номер числа s в получившейся последовательности. Если элементов последовательности, равных s , несколько, можно найти любой из их итоговых номеров.

Формат входных данных

В первой строке задано целое число n — длина последовательности ($1 \leq n \leq 100$). Во второй строке заданы целые числа a_1, a_2, \dots, a_n — элементы последовательности ($1 \leq a_i \leq 10^9$). Соседние числа в последовательности разделены пробелом.

Формат выходных данных

Выведите номер числа a_1 в отсортированной по (нестрогую) возрастанию последовательности. Если элементов последовательности, равных a_1 , несколько, можно найти любой из их итоговых номеров. Нумерация элементов начинается с единицы.

Примеры

index-after-sort.in	index-after-sort.out
4 7 8 1 5	3
3 2 2 4	2

Пояснения к примерам

В первом примере первый элемент исходной последовательности равен 7. После упорядочения последовательность принимает вид 1 5 7 8, и этот элемент становится третьим.

Во втором примере последовательность не меняет вид, а поскольку $a_1 = a_2 = 2$, подходит как ответ 1, так и ответ 2.

Задача J. Сортировка половин

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

Дана последовательность чисел чётной длины. Элементы первой половины последовательности расположите в порядке (нестрогую) возрастания, а элементы второй половины — в порядке (нестрогую) убывания. После этого выведите полученную последовательность.

Формат входных данных

В первой строке задано целое число n — длина последовательности ($1 \leq n \leq 100$, n чётно). Во второй строке заданы целые числа a_1, a_2, \dots, a_n — элементы последовательности ($1 \leq a_i \leq 10^9$). Соседние числа в последовательности разделены пробелом.

Формат выходных данных

Выведите последовательность из n чисел, которая получится, если первые $\frac{n}{2}$ элементов исходной последовательности упорядочить по возрастанию (нестрогую), а оставшиеся $\frac{n}{2}$ элементов — по убыванию (тоже нестрогую). Соседние числа в последовательности разделяйте пробелом.

Примеры

sort-halves.in	sort-halves.out
4 1 2 3 4	1 2 4 3
6 2 6 5 9 1 9	2 5 6 9 9 1

Задача K. Одинокое число

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

Дана последовательность чисел нечётной длины. Известно, что каждое число, которое встречается в этой последовательности и не равно x , встречается в ней ровно два раза, а само число x — ровно один раз. Найдите x .

Формат входных данных

В первой строке задано целое число n — длина последовательности ($1 \leq n \leq 100$, n нечётно). Во второй строке заданы целые числа a_1, a_2, \dots, a_n — элементы последовательности ($1 \leq a_i \leq 10^9$). Соседние числа в последовательности разделены пробелом. Гарантируется, что существует число x , которое встречается в последовательности ровно один раз, а все остальные элементы последовательности встречаются в ней ровно два раза.

Формат выходных данных

Выведите число x , которое встречается в заданной последовательности ровно один раз.

Примеры

lonely-number.in	lonely-number.out
3 4 7 4	7
5 2 2 5 3 5	3

Задача L. Минимальное отсутствующее число

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

Дана последовательность чисел. Найдите минимальное неотрицательное целое число, которое в ней не встречается.

Формат входных данных

В первой строке задано целое число n — длина последовательности ($1 \leq n \leq 100$). Во второй строке заданы целые числа a_1, a_2, \dots, a_n — элементы последовательности ($|a_i| \leq 10^9$). Соседние числа в последовательности разделены пробелом.

Формат выходных данных

Выведите минимальное неотрицательное целое число, которое ни разу не встречается в заданной последовательности.

Примеры

minimum-missing.in	minimum-missing.out
3 1 2 2	0
4 1 0 5 -1	2