

Задача 1. Алхимия. (100 баллов)

Имя входного файла	alchemy.in
Имя выходного файла	alchemy.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	64 мегабайт

Алхимики средневековья владели знаниями о превращении различных химических веществ друг в друга. Это подтверждают и недавние исследования археологов. В ходе археологических раскопок было обнаружено m глиняных табличек, каждая из которых была покрыта непонятными на первый взгляд символами. В результате расшифровки выяснилось, что каждая из табличек описывает одну алхимическую реакцию, которую умели проводить алхимики.

Результатом алхимической реакции является превращение одного вещества в другое. Задан набор алхимических реакций, описанных на найденных глиняных табличках, исходное вещество и требуемое вещество. Необходимо выяснить: возможно ли преобразовать исходное вещество в требуемое с помощью этого набора реакций, а в случае положительного ответа на этот вопрос — найти минимальное количество реакций, необходимое для осуществления такого преобразования.

Формат входного файла. Первая строка входного файла содержит целое число m ($0 \leq m \leq 1000$) – количество записей в книге.

Каждая из последующих m строк описывает одну алхимическую реакцию и имеет формат `вещество1 -> вещество2`, где `вещество1` – название исходного вещества, `вещество2` – название продукта алхимической реакции.

$m+2$ -ая строка входного файла содержит название вещества, которое имеется исходно, $m+3$ -ая – название вещества, которое требуется получить.

Во входном файле упоминается не более 100 различных веществ. Название каждого из веществ состоит из строчных и заглавных английских букв и имеет длину не более 20 символов. Строчные и заглавные буквы различаются.

Формат выходного файла. В выходной файл выведите минимальное количество алхимических реакций, которое требуется для получения требуемого вещества из исходного, или `-1`, если требуемое вещество невозможно получить.

Пример.

alchemy.in	alchemy.out
5 Aqua -> AquaVita AquaVita -> PhilosopherStone AquaVita -> Argentum Argentum -> Aurum AquaVita -> Aurum Aqua Aurum	2
5 Aqua -> AquaVita AquaVita -> PhilosopherStone AquaVita -> Argentum Argentum -> Aurum AquaVita -> Aurum Aqua Osmium	-1

Задача 2. Анаграммы. (100 баллов)

Имя входного файла	anagram.in
Имя выходного файла	anagram.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	64 мегабайт

Строка S_1 называется анаграммой строки S_2 , если она получается из S_2 перестановкой символов. Даны строки S_1 и S_2 . Напишите программу, которая проверяет, является ли S_1 анаграммой S_2 .

Формат входного файла. Первая строка входного файла содержит строку S_1 , вторая - S_2 . Обе строки состоят только из прописных букв английского алфавита. Строки не пусты и имеют длину не больше 100000 символов.

Формат выходного файла. В выходной файл выведите YES, если S_1 является анаграммой S_2 , и NO - в противном случае.

Пример.

anagram.in	anagram.out
АВАА	NO
АВВА	
АВВА	YES
ВАВА	

Задача 3. Coloring. (100 баллов)

Имя входного файла	coloring.in
Имя выходного файла	coloring.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	64 мегабайт

Таблицей умножения назовем таблицу размера n строк на m столбцов, в которой на пересечении i -ой строки и j -ого столбца стоит число $i \cdot j$ (строки и столбцы нумеруются с единицы).

В одной из математических школ было решено провести педагогический эксперимент. Для того, чтобы ученикам было проще запоминать таблицу умножения, некоторые числа в ней будут покрашены в красный, некоторые - в синий, а некоторые - в зеленый цвет (оставшиеся числа будут черными).

Процесс покраски чисел можно условно разбить на четыре этапа. На первом этапе все числа красятся в черный цвет. На втором - все четные числа красятся в красный цвет, на третьем - все числа, делящиеся на 3, красятся в зеленый цвет, на четвертом - все числа, делящиеся на 5, красятся в синий цвет.

Необходимо найти, сколько чисел какого цвета будет в одной раскрашенной таблице умножения n на m . Напишите программу, решающую задачу подсчета соответствующих количеств.

Формат входного файла. Входной файл содержит два натуральных числа n и m ($1 \leq n, m \leq 1000$).

Формат выходного файла. В первой строке выходного файла выведите количество чисел, покрашенных в красный цвет, во второй - в зеленый, в третьей - в синий, в четвертой - в черный. Следуйте формату, приведенному в примерах.

Пример.

coloring.in	coloring.out
10 10	RED : 21 GREEN : 39 BLUE : 36 BLACK : 4
5 2	RED : 5 GREEN : 2 BLUE : 2 BLACK : 1

Задача 4. Уравнение. (100 баллов)

Имя входного файла	equation.in
Имя выходного файла	equation.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	64 мегабайт

Напишите программу, которая сможет решить уравнение

$$ax^2 + bx + c = 0$$

при заданных коэффициентах a , b и c .

Формат входного файла. Единственная строка входного файла содержит три целых числа a , b и c , каждое из которых не превосходит по модулю 30000. Числа разделяются пробелами.

Формат выходного файла. На первой строке выходного файла выведите число корней заданного уравнения. Затем выведите сами корни по одному на строке с ошибкой, не превосходящей 0.0001. Если для заданных коэффициентов этого сделать нельзя, на единственной строке выходного файла выведите -1.

Пример.

equation.in	equation.out
1 -3 2	2 1.000000 2.000000
0 -1 6	1 6.000000

Задача 5. Сдвиг системы счисления. (100 баллов)

Имя входного файла	notation.in
Имя выходного файла	notation.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	64 мегабайт

В системе счисления по основанию b в качестве цифр выбирают числа от 0 до $b - 1$. Можно ли выбирать другие наборы цифр?

Например, вместо троичной системы счисления можно рассмотреть систему счисления, где вместо обычных цифр 0, 1, 2 есть цифры 1, 2 и 3.

Как перевести число n в эту систему счисления?

Например, число 7 в этой системе записывается как 21, так как $7 = 2 \cdot 3 + 1$, а число 22 записывается как 211, так как $22 = 2 \cdot 9 + 1 \cdot 3 + 1$.

Формат входного файла. Входной файл состоит из единственной строки, на которой записано натуральное число n , ($1 \leq n \leq 2000000000$).

Формат выходного файла. В выходной файл выведите число n записанное в указанной системе счисления.

Пример.

notation.in	notation.out
7	21
22	211

Задача 6. Сумма. (100 баллов)

Имя входного файла	sum.in
Имя выходного файла	sum.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	256 мегабайт

Во входном файле задана последовательность целых чисел. Требуется найти подпоследовательность заданной последовательности с максимальным модулем суммы входящих в нее чисел. Напомним, что модуль целого числа x равняется x , если $x \geq 0$ и $-x$, если $x < 0$.

Формат входного файла. Первая строка содержит натуральное число n ($1 \leq n \leq 10000$) - длину последовательности. Во второй строке записаны n целых чисел, по модулю не превосходящих 10000.

Формат выходного файла. В первой строке выходного файла выведите длину L выбранной вами подпоследовательности. Во второй строке должны быть записаны L различных чисел, разделенных пробелами - номера выбранных членов последовательности.

Пример.

sum.in	sum.out
5	2
-1 4 -1 6 -7	2 4

Задача 7. Поднос. (100 баллов)

Имя входного файла	tray.in
Имя выходного файла	tray.out
Максимальное время работы на одном тесте	2 секунды
Максимальный объем используемой памяти	64 мегабайт

Школьная столовая хочет заказать новые треугольные подносы. Естественно, был организован тендер. Основным критерием отбора предложений является условие того, что имеющиеся в столовой круглые тарелки должны помещаться на новые подносы. Директор школы дал вам задание написать программу, проверяющую, помещается ли тарелка данного размера на поднос с данными длинами сторон.

Формат входного файла. В первой строке входного файла находятся 3 целых положительных числа - длины сторон подноса. Гарантируется, что поднос с такими сторонами существует. Во второй строке входного файла содержится радиус тарелки - целое положительное число. Все числа во входном файле не превосходят 100. Радиус тарелки и стороны подноса указаны в одних и тех же единицах измерения.

Формат выходного файла. Выведите в выходной файл сообщение YES, если тарелка помещается на поднос, и NO - в противном случае.

Пример.

tray.in	tray.out
1 1 1	NO
1	
3 4 5	YES
1	