

Задача 1. Буквы алфавита

Источник:	базовая
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Вам дана строка, состоящая из строчных букв латинского алфавита. Все буквы в ней различны.

Требуется переставить буквы данной строки так, чтобы получившаяся строка была лексикографически больше исходной.

Поскольку в данной постановке задача слишком простая и может иметь не единственное решение, то вам требуется среди всех таких строк выбрать лексикографически минимальную.

Строка s , состоящая из символов s_0, s_1, \dots, s_n , считается *лексикографически меньше* строки t , состоящей из символов t_0, t_1, \dots, t_n , если существует индекс k такой, что:

- $s_i = t_i$ для всех $i = 0, 1, \dots, k - 1$;
- $s_k < t_k$.

Иными словами, лексикографическое сравнение строк - это привычное нам сравнение слов «по алфавиту», когда мы находим первую букву, в которой две строки различаются, и на основании этой буквы делаем вывод о том, какое из слов «меньше». Лексикографическое сравнение окружает нас повсюду: его можно найти в порядке людей в списках групп, в порядке номеров в телефонной книге, и т.д.

Формат входных данных

В первой строке входного файла записано целое число N ($2 \leq N \leq 26$) — количество символов в строке.

Во второй строке через пробел записано N строчных букв латинского алфавита. Гарантируется, что все буквы различны.

Формат выходных данных

В выходной файл необходимо вывести через пробел символы требуемой строки.

Гарантируется, что требуемая перестановка существует.

Примеры

input.txt	output.txt
5 a b c d e	a b c e d
3 q z w	w q z

Задача 2. Целое: значение - запись

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Перевести заданное целое число из десятичной системы счисления в b -ичную систему счисления.

Формат входных данных

В первой строке входного файла записано целое число b – основание системы счисления, в которую нужно перевести число ($2 \leq b \leq 16$).

Во второй строке дана десятичная запись целого числа N ($0 \leq b \leq 10^6$).

Формат выходных данных

В выходной файл необходимо вывести представление числа N в b -ичной системе счисления. Цифры, большие 9, выводить маленькими буквами латинского алфавита.

Пример

input.txt	output.txt
2	11001
25	

Задача 3. Целое: запись - значение

Источник: базовая
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Перевести заданное целое число из b -ичной в десятичную систему счисления.

Формат входных данных

В первой строке входного файла записано целое число b – основание системы счисления, в которую нужно перевести число ($2 \leq b \leq 16$).

Во второй строке дана запись целого числа N в b -ичной системе счисления ($0 \leq b \leq 10^6$). Цифры, большие 9, обозначаются маленькими буквами латинского алфавита.

Формат выходных данных

В выходной файл необходимо вывести десятичную запись целого числа – значение числа N , записанного во входном файле.

Пример

<code>input.txt</code>	<code>output.txt</code>
2 11001	25

Задача 4. Перевод целых чисел

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Дано число N в p -ичной системе счисления.

Требуется выполнить перевод числа N в q -ичную систему счисления.

Формат входных данных

В первой строке через пробел записаны три числа p , q и N ($2 \leq p, q \leq 36$). Гарантируется, что значение числа N в десятичной системе счисления не превосходит 10^9 .

Для записи цифр, значения которых в десятичной системе счисления имеют значения от 10 до 36, используются строчные латинские буквы 'a', 'b', ..., 'z'.

Формат выходных данных

Выведите число N в q -ичной системе счисления.

Примеры

input.txt	output.txt
2 16 101010	2a
7 20 22	g
20 7 g	22

Задача 5. Дробное: значение - запись

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Перевести заданное дробное число из десятичной системы счисления в b -ичную систему счисления.

Формат входных данных

В первой строке входного файла записаны два целых числа b и k – основание системы счисления, в которую нужно перевести число, и максимальное количество знаков после b -ичной точки ($2 \leq b \leq 16, 1 \leq k \leq 10$).

Во второй строке дана десятичная запись дробного числа N ($0 < N < 1$).

Формат выходных данных

В выходной файл необходимо вывести представление числа N в b -ичной системе счисления, не более чем с k знаками после точки. Выравнивающие нули не выводить. Цифры, большие 9, выводить маленькими буквами латинского алфавита.

Дробную часть вычислять с точностью до 10^{-5} .

Примеры

input.txt	output.txt
2 8 0.25	0.01
3 4 0.25	0.0202

Задача 6. Дробное: запись - значение

Источник:	основная
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Перевести заданное дробное число из b -ичной системы счисления в десятичную систему счисления.

Формат входных данных

В первой строке входного файла записано целое число b – основание системы счисления, из которой нужно перевести число ($2 \leq b \leq 16$).

Во второй строке дана b -ичная запись дробного числа N ($0 < N < 1$). Ее длина не превосходит 50. Цифры, большие 9, обозначаются маленькими буквами латинского алфавита.

Формат выходных данных

В выходной файл необходимо вывести десятичное представление вещественного числа N с пятью знаками после точки.

Примеры

input.txt	output.txt
2 0.01	0.25000
3 0.0202	0.24691

Задача 7. Перевод вещественных чисел

Источник: основная
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Перевести заданное вещественное число из b_1 -ичной системы счисления в b_2 -ичную систему счисления.

Формат входных данных

В первой строке входного файла записаны три целых числа b_1 , b_2 и k , где:

b_1 – основание системы счисления, в которой записано входное число A ,

b_2 – основание системы счисления, в которую нужно перевести число A ,

k – максимальное количество знаков после b_2 -ичной точки ($2 \leq b_1, b_2 \leq 16, 1 \leq k \leq 10$).

Во второй строке дана запись числа A в системе счисления с основанием b_1 . Число может быть и целым и вещественным. Целая часть отделяется от дробной точкой. Длина записи числа не превосходит 100. Цифры, большие 9, обозначаются маленькими буквами латинского алфавита.

Формат выходных данных

В выходной файл необходимо вывести запись числа в системе счисления с основанием b_2 . Дробную часть, если таковая имеется, выводить не более чем с k знаками после точки. Несущественные нули не выводить. Вычисления производить с точностью до 10^{-5} . Цифры, большие 9, выводить маленькими буквами латинского алфавита.

Если входное число задано некорректно, то вывести слово NO.

Примеры

input.txt	output.txt
2 4 10 11101.01101	131.122
2 3 10 11101.01201	NO
3 9 2 2102	72

Задача 8. Длинное умножение

Источник: повышенной сложности
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Во входном файле задано два целых положительных числа, одно в первой строке, второе — во второй строке. Нужно вычислить произведение этих двух чисел и вывести его в выходной файл.

Количество десятичных цифр в каждом входном числе может достигать 1 000. Поэтому требуется реализовать умножение двух длинных чисел “столбиком”.

Пример

<code>input.txt</code>	<code>output.txt</code>
1991 30457	60639887

Задача 9. Длинное деление

Источник: повышенной сложности
Имя входного файла: `input.txt`
Имя выходного файла: `output.txt`
Ограничение по времени: 1 секунда
Ограничение по памяти: разумное

Во входном файле задано два целых положительных числа, одно в первой строке, второе — во второй строке. Нужно вычислить частное этих двух чисел (округлённое вниз) и вывести его в выходной файл.

Количество десятичных цифр в каждом входном числе может достигать 1 000. Поэтому требуется реализовать деление двух длинных чисел “столбиком”.

Пример

<code>input.txt</code>	<code>output.txt</code>
1023405633 84037	12178
10000000000000000000000000000000 7	14285714285714285714285714

Задача 10. Числительные

Источник:	повышенной сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

Вася пишет программу, которая оценивает строк кода в проекте. Всё самое сложное уже написано, осталось лишь написать вывод ответа пользователю. Для красоты Вася хочет красиво распечатать результат словами, например: “две тысячи сто тридцать пять строк”. Вам нужно написать алгоритм построения этой строки.

Формат входных данных

В первой строке задано число T — количество тестовых случаев в файле ($1 \leq T \leq 10^4$).

В каждой из остальных T строк указано одно целое число N — количество строк в проекте.

Сегодня проекты стали довольно большими, поэтому диапазон: $1 \leq N < 10^9$
(<https://informationisbeautiful.net/visualizations/million-lines-of-code/>).

Формат выходных данных

В каждую строку выходного файла нужно записать словами, сколько строк в соответствующем проекте.

Чтобы избежать вопросов кодировок, слова нужно выводить транслитом. Мягкий знак заменяется на кавычку (ASCII 39).

Используется транслит BGN/PCGN. Онлайн-транслитератор есть на странице <http://www.transliteration.com/transliteration/en/russian/bgn-pcgn/>.

(пример приведён на следующей странице)

Пример

input.txt
17 1 4 9 21 33 40 54 73 173 345 797 987 1000 1986 100000 5001002 32011171
output.txt
odna stroka chetyre stroki devyat' strok dvadtsat' odna stroka tridtsat' tri stroki sorok strok pyat'desyat chetyre stroki sem'desyat tri stroki sto sem'desyat tri stroki trista sorok pyat' strok sem'sot devyanosto sem' strok devyat'sot vosem'desyat sem' strok odna tysyacha strok odna tysyacha devyat'sot vosem'desyat shest' strok sto tysyach strok pyat' millionov odna tysyacha dve stroki tridtsat' dva milliona odinnadtsat' tysyach sto sem'desyat odna stroka