Содержание

Must have 2

Задача 21A. Словарь [0.35, 256] 2

Задача 21B. K-я строка [0.1, 256] 3

Задачи здорового человека 4

Задача 21C. Телефонные номера [0.1, 256] 4

Задача 21D. Кодирование [0.1, 256] 6

Задача 21E. Декодирование [0.1, 256] 7

Для искателей острых ощущений 8

Задача 21F. Общая подстрока [0.5, 256] 8

Задача 21G. Общая подпоследовательность [3.0, 256] 9

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

Обратите внимание на `GNU C++` компиляторы с суффиксом `inc`.
Подни можно пользоваться дополнительной библиотекой (`optimization.h`).
То есть, использовать быстрый ввод-вывод: **пример про числа и строки**.
И быструю аллокацию памяти (ускоряет vector-set-map-весь-STL): пример.

Для тех, кто хочет разобраться, как всё это работает.
Короткая версия быстрого ввода-вывода (тык) и короткая версия аллокатора (тык).

Must have

Задача 21A. Словарь [0.35, 256]

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

Формат входных данных

В первой строке дан текст (не более 10^6 строчных латинских букв). Далее дано число M — количество слов в словаре.

В следующих M строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более 10^5.

Формат выходных данных

M строк вида `Yes`, если слово присутствует, и `No` иначе.

Пример

stdin	stdout
trololo 3 abacabadabacaba olo trol	No Yes Yes

Задача 21B. K-я строка [0.1, 256]

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S;
- найти в словаре k-ю строку в лексикографическом порядке.

Изначально словарь пуст.

Формат входных данных

Число команд N ($N \leqslant 10^5$). Следующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k, либо в виде строки S, которая может состоять только из строчных латинских букв. При запросе k-й строки она точно существует. Сумма длин добавляемых строк не превышает 10^5.

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k-ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5.

Примеры

stdin	stdout
7 pushkin lermontov tolstoy gogol gorkiy 5 1	tolstoy gogol

Подсказка по решению

Бор. Разобрана на практике.

Задачи здорового человека

Задача 21C. Телефонные номера [0.1, 256]

В современном мире вы встречаетесь с огромным количеством телефонных номеров, которые со временем становятся всё длиннее и длиннее. И вам приходится запоминать эти номера. Одним из простых способов запоминания является сопоставление букв каждой цифре, как показано на следующем рисунке:

1 ij	2 abc	3 def
4 gh	5 kl	6 mn
7 prs	8 tuv	9 wxy
	0 oqz	

Таким образом, каждому слову или группе слов может быть сопоставлен уникальный номер, так что можно запоминать слова вместо телефонных номеров. Очевидно, есть особый шарм в том, чтобы найти простую взаимосвязь между словом, используемым для запоминания телефонного номера, и владельцем этого номера. Так, телефонный номер 941837296 вашего друга, играющего в шахматы, может быть прочитан как WHITEPAWN (белая пешка), а номер 2855304 Вашего любимого учителя может быть прочитан как BULLDOG (бульдог). Напишите программу, находящую самую короткую последовательность слов (имеющую наименьшее количество слов), которая соответствует заданному номеру телефона и заданному списку слов. Соответствие описано на рисунке выше.

Формат входных данных

Ввод состоит из набора тестов. Первая строка каждого теста содержит номер телефона, к которому нужно подобрать мнемонику. Номер состоит не более чем из 100 цифр. Вторая строка содержит общее количество слов в словаре (максимум 50 000). Каждая из оставшихся строк содержит одно слово, состоящее не более чем из 50 строчных латинских букв. Общий размер ввода не превосходит 300 килобайт. Последняя строка ввода содержит число -1.

Формат выходных данных

Каждая строка вывода должна содержать кратчайшую последовательность слов, найденную вашей программой. Слова должны быть разделены одиночными пробелами. Если для входных данных нет решения, соответствующая строка вывода должна содержать текст “`No solution.`”. Если существует несколько решений, имеющих одинаковое количество слов, можете выбрать любое из них.

Примеры

stdin	stdout
7325189087 5 it your reality real our 4294967296 5 it your reality real our -1	reality our No solution.

Подсказка по решению

Бор.

Задача 21D. Кодирование [0.1, 256]

Известно, что в текстах практически на любом языке, имеющем алфавит, одни буквы систематически встречаются чаще, чем другие. Например, в типичном тексте на английском языке частота буквы ‘e’ больше двенадцати процентов, а частота буквы ‘z’ меньше одного процента. В русском языке буква ‘о’ имеет частоту более девяти процентов, а буква ‘ъ’ — менее одной десятой процента.

Такие различия между буквами можно использовать, чтобы хранить тексты в сжатом виде. Сопоставим каждой букве какой-то непустой двоичный код; частым буквам сопоставим коды из малого количества битов, а редким — из большого количества битов. Строку будем записывать как последовательность кодов букв в ней без разделитетей. Коды букв организуем таким образом, чтобы ни один код буквы не начинался с кода какой-то другой буквы; такие коды называются *беспрефиксными*. К примеру, если мы закодируем букву ‘e’ кодом `10`, то все остальные буквы должны иметь либо код `0`, либо код, начинающийся на `00`, `01` или `11`. Это нужно для того, чтобы в закодированном тексте не требовалось ставить разделители между буквами: при такой организации кодов любая закодированная строка восстанавливается однозначно.

От того, какие коды букв мы выберем, зависит размер закодированной строки. Например, если в строке “`abacabad`” мы закодируем буквы как “`a: 00`”, “`b: 01`”, “`c: 10`” и “`d: 11`”, то закодированная строка будет иметь вид “`00 01 00 10 00 01 00 11`” (группы цифр, соответствующие разным буквам, разделены для удобства восприятия), и размер её будет равен 16. Если же мы выберем другие беспрефиксные коды “`a: 0`”, “`b: 10`”, “`c: 110`” и “`d: 111`”, то закодированная строка будет иметь вид “`0 10 0 110 0 10 0 111`”, и её размер будет равен 14.

Дана строка s. Сопоставьте каждой её букве двоичный код так, чтобы коды букв были беспрефиксными, и при этом закодированная ими строка s имела минимальный возможный размер.

Формат входных данных

В первой строке входного файла задана непустая строка s. Она состоит из строчных букв латинского алфавита. Длина этой строки не превосходит 100 000 символов.

Формат выходных данных

В первой строке выходного файла выведите два числа k и l через пробел — количество различных букв, встречающихся в строке, и размер получившейся закодированной строки, соответственно. В следующих k строках запишите коды букв в формате “`<letter>: <code>`”. Ни один код не должен являться префиксом другого. Буквы могут быть перечислены в любом порядке. Наконец, в последней строке запишите закодированную строку.

Если ответов с минимальным значением l несколько, можно вывести любой из них.

Примеры

stdin	stdout
a	1 1 a: 0 0
abacabad	4 14 a: 0 b: 10 c: 110 d: 111 01001100100111

Задача 21E. Декодирование [0.1, 256]

В этой задаче по набору беспрефиксных кодов букв и строке, закодированной с помощью этих кодов так, как это описано в задаче **“Кодирование”**, нужно восстановить исходную строку.

Формат входных данных

В первой строке входного файла заданы два целых числа k и l через пробел — количество различных букв, встречающихся в строке, и размер получившейся закодированной строки, соответственно. В следующих k строках записаны коды букв в формате “`<letter>: <code>`”. Ни один код не является префиксом другого. Буквы могут быть перечислены в любом порядке. В качестве букв могут встречаться лишь строчные буквы латинского алфавита; каждая из этих букв встречается в строке хотя бы один раз. Наконец, в последней строке записана закодированная строка.

Исходная строка и коды всех букв непусты. Заданный код таков, что закодированная строка имеет минимальный возможный размер.

Формат выходных данных

В первой строке выходного файла выведите строку s. Она должна состоять из строчных букв латинского алфавита. Гарантируется, что длина правильного ответа не превосходит 100 000 символов.

Примеры

stdin	stdout
1 1 a: 0 0	a
4 14 a: 0 b: 10 c: 110 d: 111 01001100100111	abacabad

Для искателей острых ощущений

Задача 21F. Общая подстрока [0.5, 256]

Заданы две строки, состоящие из 0 и 1. Рассмотрим все строки, которые являются подстроками обеих данных строк. Найдите среди них k-ую в лексикографическом порядке.

Строка S меньше строки T в лексикографическом порядке, если выполняется одно из двух условий:

- S является префиксом T;
- существует i, не превышающее длин строк S и T, такое что для $j < i$ выполняется $S[j] = T[j]$ и $S[i] < T[i]$.

Формат входных данных

Первые две строки входного файла содержат заданные строки, длиной не более 4000 символов каждая. Третья строка содержит целое положительное число k, не превышающее количества общих подстрок двух заданных строк.

Формат выходных данных

Выведите в выходной файл k-ую в лексикографическом порядке общую подстроку заданных строк.

Пример

stdin	stdout
0100 0010 3	01

Задача 21G. Общая подпоследовательность [3.0, 256]

У Никиты есть массив A из N элементов.
Он хочет найти максимизировать величину

$$(A[l_1] \hat{} A[l_1+1] \hat{} \ldots \hat{} A[r_1]) + (A[l_2] \hat{} A[l_2+1] \hat{} \ldots \hat{} A[r_2])$$

где $1 \leqslant l_1 \leqslant r_1 < l_2 \leqslant r_2 \leqslant N$. Никита – обычный художник, помогите ему.

Формат входных данных

На первой строке число N ($2 \leqslant N \leqslant 10^6$). На второй строке N целых чисел от 0 до 10^9.

Формат выходных данных

Одно число – максимум.

Примеры

stdin	stdout
5 1 2 3 1 2	6