

Задача А. Частотный анализ

Имя входного файла: `frequency.in`
Имя выходного файла: `frequency.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 64 мегабайта

Дан текст. Отсортируйте все слова в нём по возрастанию количества их появления, а при одинаковой частоте появления — в лексикографическом порядке.

Формат входных данных

Во входном файле содержится исходный текст. Текст состоит не более чем из 100 000 слов, разделённых пробелами и переводами строк. Все слова состоят из строчных латинских букв. Соседние слова разделены ровно одним пробельным символом. Длина любого слова не превышает 20 символов.

Формат выходных данных

Выведите все слова, встречающиеся в тексте, по одному на каждую строку. Слова должны быть отсортированы по возрастанию их количества в тексте, а при равенстве — в лексикографическом порядке.

Примеры

<code>frequency.in</code>	<code>frequency.out</code>
<code>hi</code>	<code>james</code>
<code>hi</code>	<code>jean</code>
<code>what is your name</code>	<code>what</code>
<code>my name is bond</code>	<code>your</code>
<code>james bond</code>	<code>bond</code>
<code>my name is damme</code>	<code>claudio</code>
<code>van damme</code>	<code>hi</code>
<code>claudio van damme</code>	<code>my</code>
<code>jean claudio van damme</code>	<code>is</code>
	<code>name</code>
	<code>van</code>
	<code>damme</code>

Задача В. Встреченные ранее числа

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

Формат входных данных

Во входной строке записана последовательность из не более, чем 100 000 чисел через пробел. Каждое из чисел не превышает 10^9 по абсолютному значению.

Формат выходных данных

Для каждого числа выведите слово **YES** (в отдельной строке), если это число ранее встречалось в последовательности или **NO**, если не встречалось.

Примеры

<code>metbefore.in</code>	<code>metbefore.out</code>
1 2 3 2 3 4	NO NO NO YES YES NO

Задача С. Англо-латинский словарь

Имя входного файла: dictionary.in
Имя выходного файла: dictionary.out
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Однажды, разбирая старые книги на чердаке, школьник Вася нашёл англо-латинский словарь. Английский он к тому времени знал в совершенстве, и его мечтой было изучить латынь. Поэтому попавшийся словарь был как раз кстати.

К сожалению, для полноценного изучения языка недостаточно только одного словаря: кроме англо-латинского необходим латинско-английский. За неимением лучшего он решил сделать второй словарь из первого.

Как известно, словарь состоит из переводимых слов, к каждому из которых приводится несколько слов-переводов. Для каждого латинского слова, встречающегося где-либо в словаре, Вася предлагает найти все его переводы (то есть все английские слова, для которых наше латинское встречалось в его списке переводов), и считать их и только их переводами этого латинского слова.

Помогите Васе выполнить работу по созданию латинско-английского словаря из англо-латинского.

Формат входных данных

Во входном файле вам дан набор описаний. Каждое описание содержится в отдельной строке, в которой записано сначала английское слово, затем отведённый пробелами дефис (символ номер 45), затем разделённые запятыми с пробелами переводы этого английского слова на латинский. Переводы отсортированы в лексикографическом порядке. Порядок следования английских слов в словаре также лексикографический.

Все слова состоят только из маленьких латинских букв, длина каждого слова не превосходит 15 символов. Общее количество слов на входе не превышает 100000.

Формат выходных данных

Выведите соответствующий данному латинско-английский словарь. Выведите в первой строке количество слов в словаре, а далее - описания, в точности соблюдая формат входных данных. В частности, первым должен идти перевод лексикографически минимального латинского слова, далее — второго в этом порядке и т.д. Внутри перевода английские слова должны быть также отсортированы лексикографически.

Примеры

dictionary.in	dictionary.out
apple - malum, pomum, popula	7
fruit - baca, bacca, popum	baca - fruit
punishment - malum, multa	bacca - fruit
	malum - apple, punishment
	multa - punishment
	popum - apple
	popula - apple
	popum - fruit

Задача D. Равные подстроки

Имя входного файла: eqsubstr.in
Имя выходного файла: eqsubstr.out
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана строка $S = s_1 s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, равны ли подстроки $s_{l_1} \dots s_{r_1}$ и $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «+», если соответствующие подстроки равны, и «-», в противном случае.

Примеры

eqsubstr.in	eqsubstr.out
abacaba 4 1 1 7 7 1 3 5 7 3 4 4 5 1 7 1 7	++-+
qa 3 1 1 1 1 2 2 2 2 1 1 2 2	++-

Задача Е. Подпалиндромы

Имя входного файла: `palindrome.in`
Имя выходного файла: `palindrome.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Строка называется *палиндромом*, если она одинаково читается слева направо и справа налево. Иными словами, строка $s_1s_2\dots s_k$ — палиндром, если $s_1 = s_k$, $s_2 = s_{k-1}$ и т.д.

Дана строка $S = s_1s_2\dots s_n$ и множество запросов вида (l, r) . Для каждого запроса требуется ответить, является ли подстрока $s_l\dots s_r$ палиндромом.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l и r ($1 \leq l \leq r \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «+», если подстрока $s_l\dots s_r$ является палиндромом, и «-», в противном случае.

Примеры

<code>palindrome.in</code>	<code>palindrome.out</code>
<code>abacaba</code>	<code>++-+</code>
<code>4</code>	
<code>1 7</code>	
<code>2 2</code>	
<code>1 5</code>	
<code>3 5</code>	

Задача F. Сравнение подстрок

Имя входного файла: `substrcmp.in`
Имя выходного файла: `substrcmp.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

Дана строка $S = s_1 s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, какая из подстрок больше — $s_{l_1} \dots s_{r_1}$ или $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «=», если соответствующие подстроки равны, «>», если первая подстрока больше и «<», если первая подстрока меньше.

Примеры

substrcmp.in	substrcmp.out
abacaba	=
3	<
1 3 5 7	>
1 3 3 5	
4 7 2 5	
ab	<
2	<
1 1 2 2	
1 1 1 2	