Решения, использующие хеши, приниматься не будут!

Yandex. Яндекс

Имя входного файла: `yandex.in`
Имя выходного файла: `yandex.out`

Яндекс работает в одной очень известной компании. Его работа не очень сложна, но требует много времени. В основном Яндекс ищет некоторые данные в одной книге и переписывает их в другую. Яндекса не особенно волнует, кому нужны результаты его работы, главное — что за работу хорошо платят. Яндекс пришел в эту компанию не так давно, поэтому он трудится добросовестно и очень устает к концу дня. К концу дня для него все символы в книге сливаются, так что все эти ценные данные — это одна длинная строка, но он должен еще работать и работать с ними дальше и дальше... Может, босс заметит, как тщательно работает Яндекс, и повысит его...

Но... О, нет... Пока Яндекс мечтал, он забыл, что́ он должен был смотреть в первой книге... После перерыва и чашки чая «Липтон» он кое-что вспомнил. Во-первых, он вспомнил, что он должен был искать какую-то строку в первой книге. Во-вторых, он вспомнил, что во вторую книгу он должен был выписывать позиции, в которых встречалась эта строка, и что он уже выписал их все.

Формат входного файла

Во входном файле содержатся несколько тестов. Описание каждого теста начинается с натуральных чисел n ($1 \leqslant n \leqslant 1\,000\,000$) — количество символов в первой книге — и k ($1 \leqslant k \leqslant n$) — количество позиций, в которых Яндекс уже нашел вхождения искомой строки в текст (т. е. количество чисел во второй книге). На второй строке описания теста находится текст из первой книги — последовательность символов с ASCII-кодами, бóльшими, чем 64. Третья строка описания теста содержит k номеров позиций, которые были записаны во второй книге.

Строка с $n = k = 0$ обозначает конец тестов; этот тест и все данные после него не должны быть обработаны.

Формат выходного файла

Для каждого теста выведите в выходной файл одну строку. Если существует строка, которая входит в текст в тех и только тех позициях, что указаны во второй книге, выведите одну строку „`Correct. Length =` x`..`y`.`“, где x и y — минимально и максимально возможные длины искомой строки. Если решения не существует, выведите в выходной файл одну строку „`Mistake.`“.

Пример

yandex.in	yandex.out
5 2 ababa 1 3 1 1 a 1 2 2 ab 1 2 0 0	Correct. Length = 2..3. Correct. Length = 1..1. Mistake.

Basis. Основание строки

Имя входного файла: `basis.in`
Имя выходного файла: `basis.out`

Строка S была записана много раз подряд, после чего из получившейся строки взяли подстроку и дали вам. Ваша задача определить минимально возможную длину исходной строки S.

Формат входного файла

В первой и единственной строке входного файла записана данная вам строка. Строка содержит только латинские буквы, длина строки не превышает 50 000 символов.

Формат выходного файла

В выходной файл выведите ответ на задачу.

Пример

basis.in	basis.out
zzz	1
bcabcab	3

Prof. Жужжащий профессор

Имя входного файла: `prof.in`
Имя выходного файла: `prof.out`

В одном очень известном университете один очень известный профессор очень быстро произносил свои лекции, так, что ничего невозможно было разобрать. Студенты шутили по этому поводу, что он не говорит, а жужжит. Естественно, что про загадочного профессора никто абсолютно ничего не знал.

Но вот недавно Петя Булочкин решил предпринять исследование по изучению словарного запаса профессора. С этой целью он даже посетил одну лекцию и записал все сказанное на ней на диктофон. Затем, прокручивая дома запись с десятикратным замедлением, Петя смог записать все, что сказал профессор. Но вот незадача — профессор говорил так быстро, что, даже прослушивая замедленную запись, нельзя было точно сказать, где он делал паузы между словами. Таким образом, у Пети есть некоторый текст S, состоящий только из маленьких латинских букв — лекция, которая была прочитана профессором.

Петя решил, что те слова, которые профессор употреблял только один раз во время своей лекции, его не интересуют. Кроме того, понятно, что если профессор употреблял некоторое слово два или более раз, то существуют два неперекрывающихся вхождения этого слова в текст S. Назовем непустую строку T кандидатом в слова, если существуют два неперекрывающихся вхождения T в S. Теперь Петя хочет найти все строки, которые являются кандидатами в слова. И поможете ему в этом Вы.

Формат входного файла

Единственная строка входного файла содержит от 1 до 3000 маленьких латинских букв. Это и есть текст S, который прочитал профессор на лекции.

Формат выходного файла

Единственная строка выходного файла должна содержать одно число, равное количеству строк, являющихся кандидатами в слова.

Пример

prof.in	prof.out
bbaabbbabb	7

Invonly. Анализ строки

Имя входного файла: `invonly.in`
Имя выходного файла: `invonly.out`

Найдите количество различных непустых подстрок строки S, которые, будучи перевернутыми, встречаются в строке ровно один раз. Подстрокой называется некоторая идущая подряд последовательность символов данной строки. Две строки называются одинаковыми, если они совпадают (независимо от их местоположения в строке S). При сравнении регистр букв необходимо учитывать (т.е. '`a`' и '`A`' — это разные строки).

Формат входного файла

Во входном файле записана строка S, длина которой не меньше 1 и не больше 2000 символов. В строке S встречаются только маленькие и заглавные латинские буквы.

Формат выходного файла

В выходной файл выведите количество искомых подстрок строки S.

Пример

invonly.in	invonly.out
ABCABA	3
AaA	4

Примечание: В первом примере искомые подстроки — '`AB`', '`ABA`', '`C`'.

Cubest. Кубики — тесты

Имя входного файла: `cubes.in`
Имя выходного файла: `cubes.out`

Составление тестов по задаче «Кубики».