

Автокоррекция

Реализуйте программу, которая предлагает варианты замены слова, в котором допущена одна ошибка.

Для решения этой задачи реализуйте сжатое префиксное дерево.

Регистр букв для программы коррекции не имеет значения (слова в словаре хранятся в нижнем регистре).

Варианты ошибок - как в алгоритме Дameraу-Левенштейна: вставка лишнего символа, удаление символа, замена символа или транспозиция соседних символов.

Реализация алгоритма должна быть инкапсулирована. В комментариях напишите сложность ключевых алгоритмов с пояснением.

Обход детей узла можно и нужно реализовать в среднем за время, линейно зависящее от длины подходящего префикса. Соответственно, проверка наличия слова в префиксном дереве — это в среднем линейная операция, зависящая только от длины слова.

Формат ввода

Данные подаются на стандартный поток ввода. Пустые строки игнорируются.

Первая строка содержит число N - количество слов в словаре.

Последующие N строк содержат слова из словаря, по одному в строке.

Остальные строки - слова, которые надо проверять.

Формат вывода

Каждая строка выхода содержит предложение для исправления слова. Слова обрабатываются в порядке их появления.

Если слово не содержит ошибок, то выводится "%слово% - ok".

Если слово содержит одну ошибку, то выводится "%слово% -> %слово_в_словаре%". Если вариантов несколько, то они сортируются лексикографически и разделяются запятой с пробелом.

Если слово содержит более одной ошибки, то выводится "%слово% -?"

Результат работы программы выводится в стандартный поток вывода.

Пример

Ввод:

some
random
words
for
testing
your
solutions
far

some
randoms
wards
seeking
fro
solution
fur

Вывод:

some - ok
randoms -> random
wards -> words
seeking -?
fro -> for
solution -> solutions
fur -> far, for