

## Задача А. Автодополнение

Имя входного файла:	tabs.in
Имя выходного файла:	tabs.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 Мб

Вы участвуете в разработке новой оболочки командной строки tcs-ng “Tab Completing Shell — Next Generation”. Ключевой особенностью вашей оболочки будет динамический словарь команд: вместо поиска по директориям, оболочка будет иметь свой собственный словарь который может быть изменен пользователем с помощью встроенных команд “add” и “remove”. Изначально словарь содержит только эти две команды и они не могут быть удалены пользователем.

Вам дается последовательность команд введенных в оболочке. Каждая имеет вид `<command> <argument> <argument> ... <argument>` где `<command>` и `<argument>` это последовательности символов, состоящих только из строчных латинских букв и знаков “\*”. Для каждой строки ваша программа должна выполнять следующие операции.

Во-первых, каждый знак “\*” во входном файле должен быть заменен последовательностью букв (возможно пустой). Рассмотрим часть слова, которая была построена до появления знака “\*”. Обозначим ее *S*. Если:

1. никакая команда не совпадает с *S*
2. существует непустое множество команд в словаре, начинающихся с *S* и все эти команды имеют одинаковую следующую букву

то вы должны добавить эту букву к текущему слову. После чего перейти к пункту 1, то есть продолжить автодополнение. Автодополнение должно закончиться если одно из условий 1 или 2 не выполняется.

Во-вторых, строка, полученная после замены знаков “\*” должна быть обработана как команда. Для каждой команды должен быть выведен результат работы.

Если дана команда “add”, то оболочка должна выполнить следующее:

- Добавить аргументы в словарь в порядке слева направо
- Если список аргументов пуст, то выведите  
`%Operation has no result: add`
- Если аргумент уже содержится в словаре, то вся команда “add” должна быть отменена и необходимо вывести

`%Unable to add a word that already exists: <word>`

где `<word>` это первое слово, которое не может быть добавлено.

- Иначе выведите сообщение

`%Added the following words: <word> <word> ... <word>`

где `<word>` это слова, которые были добавлены в том же порядке как и в команде “add”

Если дана команда “remove”, то оболочка должна выполнить следующее:

- Удалить все аргументы из словаря в порядке слева направо.

- Если список аргументов пуст, то выведите

`%Operation has no result: remove`

- Если один из аргументов это “add” или “remove”, то вся команда “remove” должна быть отменена и необходимо вывести  
`%Such a destructive operation is not permitted.`

- Если удаляемый аргумент не содержится в словаре, то вся команда “remove” должна быть отменена и необходимо вывести

`%Unable to remove a word that does not exist: <word>`

где `<word>` это первое слово, которое не может быть удалено.

- Иначе выведите сообщение

`%Removed the following words: <word> <word> ... <word>`

где `<word>` это слова, которые были удалены в том же порядке как и в команде “remove”

Если дана любая другая команда, то необходимо вывести ее.

### Формат входного файла

Входной файл содержит команды по одной на строке. Каждое слово `<command>` и `<argument>` содержит не более 20 символов, включая “\*”. Каждая строка содержит не более 250 символов. Общий размер входного файла не превышает 256 Кб.

### Формат выходного файла

В выходной файл выведите результат работы.

## Пример

tabs.in	tabs.out
add tcsng	%Added the following words: tcsng
t*	tcsng
ad* tcs	%Added the following words: tcs
t*	tcs
t*n*	tcsng
remove tcsng	%Removed the following words: tcsng
t*	tcs
r*** tcs	%Removed the following words: tcs
t*	t
add add	%Unable to add a word that already exists: add
remove remove	%Such a destructive operation is not permitted.
add bad bat bad	%Unable to add a word that already exists: bad
b*	b
add search	%Added the following words: search
add s*desc	%Added the following words: searchdesc
s*d*	searchdesc
stop	stop

## Задача В. Prof. Жужжащий профессор

Имя входного файла: prof.in  
Имя выходного файла: prof.out  
Ограничение по времени: 0.1 секунды  
Ограничение по памяти: 16 мегабайт

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

## Задача С. Подстроки

Имя входного файла: substr.in  
Имя выходного файла: substr.out  
Ограничение по времени: 0.4 секунды  
Ограничение по памяти: 16 мегабайт

Дана строка  $s$ . Вам требуется подсчитать количество её различных подстрок. Пустую строку учитывать не следует.

### Формат входного файла

В единственной строке входного файла содержится данная строка  $s$ , состоящая из строчных латинских букв. Длина строки не превосходит 20 000 символов.

### Формат выходного файла

В единственной строке выходного файла выведите единственное число — количество различных подстрок  $s$ .

substr.in	substr.out
aaaa	4
abacaba	21