

## Задача А. Циклические суффиксы

Имя входного файла: cyclic.in  
Имя выходного файла: cyclic.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Рассмотрим строку  $S = s_1s_2s_3 \dots s_{n-1}s_n$  над алфавитом  $\Sigma$ . *Циклическим расширением* порядка  $t$  строки  $S$  назовем строку  $s_1s_2s_3 \dots s_{n-1}s_ns_1s_2 \dots$  из  $t$  символов; это значит, что мы приписываем строку  $S$  саму к себе, пока не получим требуемую длину, и берем префикс длины  $t$ .

*Циклической строкой*  $\tilde{S}$  назовем бесконечное циклическое расширение строки  $S$ .

Рассмотрим суффиксы циклической строки  $\tilde{S}$ . Очевидно, существует не более  $|S|$  различных суффиксов:  $(n + 1)$ -ый суффикс совпадает с первым,  $(n + 2)$ -ой совпадает со вторым, и так далее. Более того, различных суффиксов может быть даже меньше. Например, если  $S = abab$ , первые четыре суффикса циклической строки  $\tilde{S}$  — это:

$$\begin{aligned}\tilde{S}_1 &= abababab\dots \\ \tilde{S}_2 &= babababa\dots \\ \tilde{S}_3 &= abababab\dots \\ \tilde{S}_4 &= babababa\dots\end{aligned}$$

Здесь существует всего два различных суффикса, в то время как  $|S| = 4$ .

Отсортируем первые  $|S|$  суффиксов  $\tilde{S}$  лексикографически. Если два суффикса совпадают, первым поставим суффикс с меньшим индексом. Теперь нас интересует следующий вопрос: на каком месте в этом списке стоит сама строка  $\tilde{S}$ ?

Например, рассмотрим строку  $S = cabcab$ :

$$\begin{aligned}(1) \quad \tilde{S}_2 &= abcabcabca\dots \\ (2) \quad \tilde{S}_5 &= abcabcabca\dots \\ (3) \quad \tilde{S}_3 &= bcabcabcab\dots \\ (4) \quad \tilde{S}_6 &= bcabcabcab\dots \\ (5) \quad \tilde{S}_1 &= cabcabcab\dots \\ (6) \quad \tilde{S}_4 &= cabcabcab\dots\end{aligned}$$

Здесь циклическая строка  $\tilde{S} = \tilde{S}_1$  находится на пятом месте.

Вам дана строка  $S$ . Ваша задача — найти позицию циклической строки  $\tilde{S}$  в описанном порядке.

## Формат входного файла

Во входном файле записана единственная строка  $S$  ( $1 \leq |S| \leq 1\,000\,000$ ), состоящая из прописных латинских букв.

## Формат выходного файла

В выходной файл выведите единственное число — номер строки  $\tilde{S}$  в описанном порядке среди первых  $|S|$  суффиксов.

## Примеры

cyclic.in	cyclic.out
abracadabra	3
cabcab	5

## Задача В. Ненокку

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

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой не осторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

## Формат входного файла

В каждой строчке входного файла записано одна из двух записей.

- ? <слово> (<слово> - это набор не более 50 латинских символов);
- A <текст> (<текст> - это набор не более  $10^5$  латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведении.

2 означает добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более  $10^5$  символов. А входной файл содержит не более 15 мегабайт информации.

### Формат выходного файла

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

### Пример

nenokku.in	nenokku.out
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

### Задача С. Палиндромы

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

Сотрудник НИИ Исследования Данных Строк (НИИИДС) Вася Пупкин изучает свойства палиндромов. Напомним, что строка  $s$  длины  $n$  называется палиндромом, если она равна  $s^R$  (развёрнутой строке  $s$ ). Вася занялся подсчётом количества подстрок данной строки, являющихся палиндромами. С этой задачей Вася справился достаточно быстро. Теперь Вася решил поочерёдно заменять символы в строке на символ ‘?’, который Вася считает равным любому символу. То есть, например, Вася считает равными строки “ABA” и “A?A”, “ABV” и “AB?”, а, соответственно, строки “AB?”, “ABC??”, “?VSV?” он считает палиндромами. Вася хочет после каждой замены символа на ‘?’ определять количество подстрок строки  $s$ , являющихся палиндромами (в том смысле, в котором это понимает Вася). При этом подстроки Вася считает различными, если у них различаются позиции начала и/или конца. Вам предложено автоматизировать Васиные действия, чтобы он смог заняться чем-нибудь ещё.

### Формат входного файла

Первая строка входных данных содержит строку  $s$ , состоящую только из строчных латинских букв. Гарантируется, что строка  $s$  непуста, и её длина  $n$  не превосходит 4 000. Далее во входных данных содержится  $n$  чисел от 1 до  $n$  — номера символов, заменяемых на ‘?’.

После прочтения строки  $s$  выведите количество её подстрок, являющихся палиндромами.

Далее Вы должны  $n$  раз проделать следующую последовательность действий:

- считать число  $k$  — номер символа строки, заменённого на ‘?’ на текущем шаге ( $1 \leq k \leq n$ ),
- подсчитать количество подстрок текущей строки, являющихся палиндромами,
- вывести это число на отдельной строке,
- выполнить команду `flush(output)` в Паскале и `fflush(stdout)` в C и C++.

Гарантируется, что никакой символ исходной строки не будет дважды заменён на ‘?’.

Не рекомендуется использовать потоки ввода/вывода C++ (`fstream`), так как они работают существенно медленней `FILE *` и могут стать причиной превышения ограничения по времени.

### Формат выходного файла

Смотрите описание входных данных.

### Пример

palindromes.in	palindromes.out
abac	5
3	7
2	9
1	10
4	10

Комментарий к примеру: В строке `abac` палиндромами являются подстроки  $(1,1)$ ,  $(2,2)$ ,  $(3,3)$ ,  $(4,4)$ ,  $(1,3)$ . После замены третьего символа на ? получаем строку “ab?c”, в которой палиндромами являются подстроки  $(1,1)$ ,  $(2,2)$ ,  $(3,3)$ ,  $(4,4)$ ,  $(1,3)$ ,  $(2,3)$ ,  $(3,4)$ . После замены в ней второго символа получаем строку “a??c”, в которой палиндромом не является только подстрока  $(1,4)$ . После следующих замен все подстроки будут являться палиндромами.