

Задача А. Наивный поиск подстроки в строке

Имя входного файла: `search1.in`
Имя выходного файла: `search1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^4$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

search1.in	search1.out
aba	2
abaCaba	1 5

Задача В. Быстрый поиск подстроки в строке

Имя входного файла: `search2.in`
Имя выходного файла: `search2.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

search2.in	search2.out
aba	2
abaCaba	1 5

Задача С. Префикс-функция

Имя входного файла: `prefix.in`
Имя выходного файла: `prefix.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Постройте префикс-функцию для заданной строки s .

Формат входных данных

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

Формат выходных данных

Выведите значения префикс-функции строки s для всех индексов $1, 2, \dots, |s|$.

Примеры

<code>prefix.in</code>	<code>prefix.out</code>
aaaAAA	0 1 2 0 0 0
abacaba	0 0 1 0 1 2 3

Задача D. Z-функция

Имя входного файла: `z.in`
Имя выходного файла: `z.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Постройте Z-функцию для заданной строки s .

Формат входных данных

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

Формат выходных данных

Выведите значения Z-функции строки s для индексов $2, 3, \dots, |s|$.

Примеры

<code>z.in</code>	<code>z.out</code>
aaaAAA	2 1 0 0 0
abacaba	0 1 0 3 0 1

Задача Е. Поиск подстроки в строке с одним несовпадением

Имя входного файла: search3.in
Имя выходного файла: search3.out
Ограничение по времени: 2 секунды
Ограничение по памяти:

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки, с точностью до возможного несовпадения одного символа.

Формат входных данных

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Формат выходных данных

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Примеры

search3.in	search3.out
aaaa	4
Caaabdaaaa	1 2 6 7

Задача F. Поиск периода

Имя входного файла: `period.in`
Имя выходного файла: `period.out`
Ограничение по времени: 2 секунды
Ограничение по памяти:

Дана строка s . Требуется найти минимальную по длине строку t , такую что s представима в виде конкатенации одной или нескольких строк t .

Формат входных данных

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

Формат выходных данных

Выведите длину искомой строки t .

Примеры

<code>period.in</code>	<code>period.out</code>
abcbabcabc	3
abacaba	7

Задача G. Множественный поиск

Имя входного файла: `search4.in`
Имя выходного файла: `search4.out`
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Дано множество строк S и строка t . Требуется для каждой строки $p \in S$ определить, встречается ли она в t как подстрока.

Формат входных данных

Первая строка входного файла содержит целое число n — мощность S ($1 \leq n \leq 10^6$). Следующие n строк содержат по одной строке из S . Сумма длин всех строк из S не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$). Все строки состоят из строчных латинских букв.

Формат выходных данных

Для каждой строки из S выведите «YES», если она встречается в t и «NO» в противном случае. Строки нумеруются в порядке появления во входном файле.

Примеры

<code>search4.in</code>	<code>search4.out</code>
3	YES
abc	NO
abcdr	YES
abcde	
xabcdef	

Задача Н. Сравнения подстрок

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

Дана строка. Нужно уметь отвечать на запросы вида: равны ли подстроки $[a..b]$ и $[c..d]$.

Формат входных данных

Сперва строка S (не более 10^5 строчных латинских букв). Далее число M — количество запросов.

В следующих M строках запросы a, b, c, d . $0 \leq M \leq 10^5, 1 \leq a \leq b \leq |S|, 1 \leq c \leq d \leq |S|$

Формат выходных данных

M строк. Выведите **Yes**, если подстроки совпадают, и **No** иначе.

Пример

substrcmp.in	substrcmp.out
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

Задача I. Словарь

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

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

Формат входных данных

В первой строке дан текст (не более 10^6 строчных латинских букв). Далее дано число M — количество слов в словаре.

В следующих M строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более 10^5 .

Формат выходных данных

M строк вида `Yes`, если слово присутствует, и `No` иначе.

Пример

<code>dictionary.in</code>	<code>dictionary.out</code>
<code>trololo</code>	<code>No</code>
<code>3</code>	<code>Yes</code>
<code>abacabadabacaba</code>	<code>Yes</code>
<code>olo</code>	
<code>trol</code>	

Задача J. Помогите, спасите!

Имя входного файла: `keepcounted.in`
Имя выходного файла: `keepcounted.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 16 мегабайт

Дана строка. Найдите для каждого её префикса количество различных подстрок в нём.

Формат входных данных

В единственной строке входных данных содержится непустая строка S , состоящая из N ($1 \leq N \leq 10^4$) маленьких букв английского алфавита.

Формат выходных данных

Выведите N строк, в i -й строке должно содержаться количество различных подстрок в i -м префиксе строки S .

Примеры

<code>keepcounted.in</code>	<code>keepcounted.out</code>
<code>aabab</code>	1
	2
	5
	8
	11
<code>atari</code>	1
	3
	5
	9
	14

Задача К. Запросы на строках

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 512 мегабайт

Дана строка s , состоящая из строчных букв латинского алфавита, обозначим $f(s)$, как количество различных подстрок в s .

Вы должны уметь отвечать на запрос $f(s[l..r])$, где $s[l..r]$ означает подстроку, начинающуюся в l и заканчивающуюся в r , все границы включительно.

Формат входных данных

Первая строка содержит строку s , состоящую из строчных букв латинского алфавита ($1 \leq |s| \leq 5000$). Строка строка содержит число q — количество запросов ($1 \leq q \leq 10^4$). Следующие q строк содержат по два числа l и r — границы запросов ($1 \leq l \leq r \leq |s|$).

Формат выходных данных

Для каждого запроса выведите ответ на него.

Примеры

стандартный ввод	стандартный вывод
bbaba	3
5	1
3 4	7
2 2	5
2 5	8
2 4	
1 4	
baaba	1
5	3
3 3	8
3 4	5
1 4	1
3 5	
5 5	

Задача L. Curiosity

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

Curiosity — это марсоход, который исследует кратер «Гейл» на Марсе. Не так давно, он нашел следы воды в марсианской почве, что в будущем упростит планирование миссий для людей.

Curiosity может напрямую контактировать с Землей на скорости до 32 килобит в секунду, но в среднем нужно 14 минут и 6 секунд, чтобы сигнал дошел от Земли до Марса.

«Вы только что увидели камень и нажали на тормоз, но вы уже знаете, что вездеход уже проходит этот камень» — объясняет Мэтт Хэверли, водитель марсохода. «Так мы только планируем путь, а потом записываем набор простых текстовых команд: поехать на один метр вперед, повернуть налево, сделать фото и т.д.».

Иногда требуется реагировать очень быстро на неожиданные события. Например, если камеры увидели что-то интересное, тогда вы могли бы захотеть изменить маршрут марсохода, чтобы он сделал еще одно фото. Чтобы сделать это, вы отправляете команду замены в форме `s/⟨string⟩/⟨replacement⟩/g`. Это заменяет все вхождения `⟨string⟩`, начиная с самого левого, на `⟨replacement⟩`.

Более формально, если A — непустая строка и B — строка, тогда чтобы применить замену `s/A/B/g` к строке S , нужно сделать следующее:

- Найти самое левое вхождение строки A в S , такое, что $S = S_L + A + S_R$.
- Если нет ни одного вхождения, то ответом является S .
- Пусть R — результат применения `s/A/B/g` к строке S_R .
- Ответом является $S_L + B + R$.

Это означает, что:

- Если существует два пересекающихся вхождения строки A в S , только левое из них заменяется. Например, если применить `s/aba/c/g` к `abababa`, получим `cbc`: после замены первого вхождения `aba` строка превратится в `cbaba`, и только последнее вхождение `aba` может быть заменено после этого.
- Никакая замена не может использовать результат предыдущих замен. Например, применив `s/a/ab/g` к `a`, получим `ab`, а, применив `s/a/ba/g` к `a`, получим `ba`.

Вы знаете, что чем длиннее команда, тем больше времени занимает ее отправка. Тем самым, от вас требуется написать программу, которая найдет кратчайшую команду, изменяющую начальную строку в конечную.

Формат входных данных

Первые две строки содержат начальную и конечную строки, соответственно.

Обе строки не пусты и их длины не превосходят 2000 символов.

Строка содержит только латинские буквы, пробелы и знаки пунктуации (запятые, двоеточия, точки с запятой и дефисы).

Заданные строки не равны друг другу.

Формат выходных данных

Выведите команду, которая изменяет начальную строку в конечную минимальной длины. Если существует несколько кратчайших команд, выведите любую.

Примеры

curiosity.in	curiosity.out
move left, move right; move up move left, move down, move up	s/right;/down,/g
If not found: move x; else move -x If found: move x; else move -x	s/ not//g
abababa cbc	s/aba/c/g

Задача М. Цензура

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

Посчитайте, сколько строк над алфавитом из n символов длины m не содержат ни одной подстроки из заданного множества “запрещенных” строк.

Формат входных данных

В первой строке написаны целые числа n ($1 \leq n \leq 100$) — количество символов в алфавите, m ($1 \leq m \leq 100$) — длина искоемых строк и p ($0 \leq p \leq 10$) — количество “запрещенных” подстрок. Следующая строка содержит n символов с кодами больше 32 — буквы алфавита. Далее идет p “запрещенных” строк, длины которых не превосходят $\min(m, 10)$ символов. Строки целиком состоят из символов алфавита.

Формат выходных данных

В первой строке выведите ответ на задачу.

Пример

<code>censored.in</code>	<code>censored.out</code>
2 3 1 ab bb	5

Задача N. Подозрительные строки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вы работаете в компании, специализирующейся на интернет-технологиях, и сейчас вы разрабатываете спам-фильтр. Этот фильтр определяет, содержит ли строка спам, для этого он использует словарь «спам-слов». Если входная строка содержит по крайней мере одно слово из этого словаря в качестве подстроки, то фильтр считает, что это сообщение подозрительное. (примечание: вся строка считается подстрокой для самой себя)

Вы решили решить более сложную задачу: посчитать, сколько существует различных строк длины l , состоящих из строчных букв, который являются подозрительными для данного фильтра. Выведите ответ по модулю 10000.

Формат входных данных

В первой строке записано число n — число спам-слов в словаре ($1 \leq n \leq 10$). Во второй строке перечислены спам-слова через пробел (длина строк не больше 10, строки состоят только из строчных латинских букв). В третьей строке задано число l ($1 \leq l \leq 2^{31} - 1$).

Формат выходных данных

Выведите число подозрительных строк длины l по модулю 10000.

Примеры

стандартный ввод	стандартный вывод
1 x 1	1
2 ab bb 2	2
2 ab bb 5	6350
2 aab bba 5	4054
9 xxxxxx xxx x уухуу xxxуххх у ух ху zzzzzzzzzz 5	8752
10 aaaaaaaaa bbbbbbbbbb cccccccccc dddddddddd eeeeeeeeeee ffffffff gggggggggg hhhhhhhhhh xxxxxxxxxx zzzzzzzzzz 2147483647	5040

Задача О. Варвары и Тарзан

Имя входного файла: `divljak.in`
Имя выходного файла: `divljak.out`
Ограничение по времени: 6 секунд
Ограничение по памяти: 1024 мегабайта

Задача про странных людей — варваров. Их много, но только некоторые из них действительно важны. В этой истории есть n важных варваров, которые пронумерованы числами от 1 до n . У каждого из них есть камень, на котором написано слово, состоящее только из строчных латинских букв.

Наши варвары играют в интересную игру с их другом Тарзаном.

Игра проходит в q раундов. Всего есть два типа раундов, каждый из них определяется Тарзаном:

1. Тарзан показывает слово p варварам
2. Тарзан задает варвару с номером s следующий вопрос: «Среди всех слов, которые я тебе показывал, сколько из них содержат слово на твоём камне как подстроку?»

Помогите варварам ответить на вопросы Тарзана правильно.

Формат входных данных

Первая строка содержит целое число n ($1 \leq n \leq 10^5$) — число варваров.

В каждой из следующих n строк содержится слово из строчных букв латинского алфавита, i -е из них — слово на камне у варвара с номером i .

Далее следует целое число q ($1 \leq q \leq 10^5$) — число раундов в игре.

Следующие q строк описывают раунды игры: i -я строка описывает i -й раунд игры. Каждая строка содержит число z . Если z равно 1, то это раунд первого типа, и далее в той же строке содержится число p , состоящее из строчных букв латинского алфавита.

Если же z равно 2, то это раунд второго типа, и в той же строке содержится число s ($1 \leq s \leq n$) — номер варвара, которому Тарзан задал вопрос.

Суммарная длина всех слов, записанных на камнях у варваров, не превосходит $2 \cdot 10^6$. Суммарная длина всех слов, которые Тарзан показывает варварам, не превосходит $2 \cdot 10^6$.

Формат выходных данных

Для каждого раунда второго типа выведите ответ на вопрос Тарзана.

Примеры

divljak.in	divljak.out
3 a bc abc 3 1 abca 2 1 2 3	1 1
7 abba bbaa b bbaa abba a ba 7 1 aaabbabbaab 2 7 1 baabaaa 1 aabbbab 2 3 1 aabba 2 3	1 3 4

Замечание

Пояснение к примеру

Пояснение к первому примеру: Единственное слово, которое показал Тарзан, это **abca**. Ответ на первый вопрос — 1, потому что слово **a** является подстрокой слова **abca**. Ответ на второй вопрос — также 1, потому что слово **abc** является подстрокой слова **abca**.

Задача Р. Обратная задача Префикс-функции

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 0.5 секунд
Ограничение по памяти: 256 мегабайт

Боб только что выучил алгоритм Кнута-Мориса-Пратта.

Для строки $S = s_1s_2 \dots s_n$, $\pi(S) = (f_2, f_3, \dots, f_n)$, где f_i это максимальное $j < i$ такое, что $s_1s_2 \dots s_j = s_{i-j+1}s_{i-j+2} \dots s_i$.

По заданным f_2, f_3, \dots, f_n и размеру алфавита найдите количество строк S таких, что $\pi(S) = (f_2, f_3, \dots, f_n)$. Так как ответ может быть слишком большим, выведите его по модулю $(10^9 + 7)$.

Формат входных данных

Первая строка входа содержит два целых числа n и c , задающие длину строки и размера алфавита, соответственно ($2 \leq n \leq 2 \cdot 10^5, 1 \leq c \leq 10^9$).

Вторая строка содержит $(n - 1)$ целых чисел f_2, f_3, \dots, f_n ($0 \leq f_i < i$).

Гарантируется, что количество требуемых строк непусто.

Формат выходных данных

Выведите одно целое число — остаток от деления количества подходящих строк по модулю $10^9 + 7$

Примеры

стандартный ввод	стандартный вывод
3 3 0 0	12
5 10000000000 1 2 3 4	10000000000