

## Задача А. Суффиксный автомат

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дана строка. Постройте её суффиксный автомат.

### Формат входных данных

Строка длины от 1 до 100 000, состоящая из строчных латинских букв.

### Формат выходных данных

На первой строке число состояний автомата и число рёбер. Следующие строки содержат рёбра в формате «откуда» «куда» «символ на ребре». Далее число терминальных состояний и строка, содержащая все терминальные состояния в произвольном порядке. Начальным состоянием автомата должно быть состояние номер один.

### Пример

стандартный ввод	стандартный вывод
ababb	7 9 1 2 a 1 7 b 2 3 b 3 4 a 3 6 b 4 5 b 5 6 b 7 4 a 7 6 b 3 6 7 1

## Задача В. Ненокку

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
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 неосторожности 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 входных данных

В каждой строчке входного файла записано одна из двух записей.

1. ? <слово> (<слово> — это набор не более 50 латинских символов);
2. 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. Суммарная длина всех запросов не превосходит 15 мегабайт плюс 12140 байт.

### Формат выходных данных

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

### Пример

стандартный ввод	стандартный вывод
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

## Задача С. Помогите, спасите!

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 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  $n$  ( $1 \leq n \leq 2 \cdot 10^5$ ) строчных 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

стандартный ввод	стандартный вывод
aabab	1 2 5 8 11
atari	1 3 5 9 14

## Задача D. Конкатенация

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дана строка  $s$ , состоящая из строчных букв латинского алфавита. Рассмотрим строку  $t$ , представляющую собой конкатенацию всех подстрок  $s$  в лексикографическом порядке.

Например, если  $s = aba$ , ее подстроки — это  $\{a, b, a, ab, ba, aba\}$ , подстроки в лексикографическом порядке — это  $\{a, a, ab, aba, b, ba\}$ , и таким образом,  $t = aaabababba$ .

Также даны  $m$  запросов  $a_1, \dots, a_m$ . Для  $i$ -го запроса ответом служит  $a_i$ -й символ строки  $t$ .

### Формат входных данных

Ввод состоит из одного или более тестов.

Каждый тест начинается строкой, содержащей натуральное число  $m$ , задающим число запросов. Следующая строка содержит строку  $s$  ( $1 \leq |s| \leq 5000$ ). Следующая строка содержит  $m$  целых чисел  $a_i$  ( $1 \leq a_i \leq |t|$ ), задающих запросы.

Ввод будет завершён тестом с  $m = 0$ , который не требуется обрабатывать.

Сумма  $m$  по всем тестам во вводе не превысит 5000.

Сумма длин всех строк  $s$  также не превысит 5000.

### Формат выходных данных

Для каждого теста выведите строку из  $m$  символов: ответы на запросы.

### Пример

стандартный ввод	стандартный вывод
10	aaabababba
aba	x
1 2 3 4 5 6 7 8 9 10	
1	
x	
1	
0	

## Задача E. LZSS encoding

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Алиса хочет отправить сообщение Бобу. Она хочет зашифровать сообщение, используя оригинальный метод шифрования. Сообщение – строка  $s$ , состоящая из  $n$  строчных английских букв. В этой задаче предполагается 0-индексация строки  $s$ .

Если первые  $i$  букв сообщения уже зашифрованы, то Алиса найдет максимально возможное  $k \geq 0$ , для которого найдется индекс  $j < i$  такой, что  $s[j \dots j + k] = s[i \dots i + k]$ .

Если требуемое  $k$  нашлось, то Алиса выберет минимальный подходящий индекс  $j$  и запишет пару чисел  $(k, j)$  в шифр. Иначе Алиса добавит в шифр пару  $(-1, x_i)$ , где  $x_i$  – это ASCII-код символа  $s_i$ .

Помогите Алисе реализовать ее метод шифрования.

### Формат входных данных

Первая строка ввода содержит количество наборов входных данных  $t$  ( $1 \leq t \leq 50$ ).

Каждый набор данных описывается единственной строкой  $s$  ( $1 \leq |s| \leq 10^5$ ) – сообщением для шифровки.

Все сообщения состоят из строчных английских букв. Также гарантируется что суммарная длина  $s$  по всем наборам входных данных не превосходит  $2 \cdot 10^6$ .

### Формат выходных данных

Для каждого набора входных данных на отдельной строке выведите «Case #X:», где  $X$  – номер теста.

Далее выведите шифр, в каждой строке по два целых числа через пробел.

### Пример

стандартный ввод	стандартный вывод
3	Case #1:
aaaaaa	-1 97
aaaaabbbbbaaabbc	5 0
aba	Case #2:
	-1 97
	4 0
	-1 98
	4 5
	5 2
	-1 99
	Case #3:
	-1 97
	-1 98
	1 0

## Задача F. Ключ к шифру

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Сейчас Эркюль Пуаро занят разоблачением международного преступного синдиката, занимающегося контрабандой предметов искусства. Полиция, сотрудничающая с Пуаро, перехватила зашифрованное письмо, содержащее информацию о месте и времени предстоящей сделки, на которой будет присутствовать и глава синдиката. Чтобы сорвать сделку и задержать главу синдиката, необходимо расшифровать перехваченное письмо.

Эркюль знает, что ключ к шифру вычисляется из строки  $s$ . Обозначим за  $f(w)$  длину максимального суффикса  $w$ , не равного  $w$ , который является и префиксом  $w$ . Например,  $f(abc) = 0$ ,  $f(abab) = 2$ ,  $f(aaa) = 2$ . Тогда ключом является максимум по всем  $t$ , являющимся подстроками  $s$ , величины  $(|t| + f(t)^2)$ . Помогите Эркюлю вычислить ключ.

### Формат входных данных

В единственной строке дана строка  $s$ , состоящая из строчных латинских букв ( $1 \leq |s| \leq 500\,000$ ).

### Формат выходных данных

Выведите единственное целое число — искомый ключ к шифру.

### Пример

стандартный ввод	стандартный вывод
ababaab	14

## Задача G. Пестрая лента

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7 секунд
Ограничение по памяти:	512 мегабайт

Ильдар взял ленточку и покрасил её в несколько цветов. Формально, ленточка разбита на  $n$  клеток, каждую из которых он покрасил в один из 26 цветов, которые можно обозначить строчными буквами латинского алфавита.

Ильдар решил, что он выберет отрезок ленточки  $[l, r]$  ( $1 \leq l \leq r \leq n$ ), который ему нравится, и вырежет его из ленточки. Таким образом получится новая ленточка, которую можно условно представить как строку  $t = s_l s_{l+1} \dots s_r$ .

Теперь Ильдар играет в следующую игру — разрезает ленточку  $t$  на несколько новых ленточек и смотрит, сколько среди них различных. Формально, Ильдар выбирает  $1 \leq k \leq |t|$  индексов  $1 \leq i_1 < i_2 < \dots < i_k = |t|$  и разрезает  $t$  на  $k$  ленточек-строк  $t_1 t_2 \dots t_{i_1}, t_{i_1+1} \dots t_{i_2}, \dots, t_{i_{k-1}+1} \dots t_{i_k}$  и считает среди них количество различных. Внезапно Ильдара заинтересовал вопрос — а какое минимальное количество различных как строк из них может получиться, при условии что среди этих кусочков хотя бы один вид ленточек повторяется хотя бы два раза? Результатом игры Ильдар считает это число. Если не существует ни одного способа разрезать  $t$  таким образом, то результатом игры будет число  $-1$ .

К сожалению, Ильдар ещё не выбрал, какой именно отрезок ему нравится, но у него есть  $q$  отрезков-предпочтений  $[l_1, r_1], [l_2, r_2], \dots, [l_q, r_q]$ . Посчитайте для каждого из отрезков какой будет результат игры нём.

### Формат входных данных

Первая строка входных данных содержит одно целое число  $n$  ( $1 \leq n \leq 200\,000$ ) — длина изначальной ленточки у Ильдара.

Вторая строка содержит строку  $s$ , состоящую из  $n$  строчных букв латинского алфавита — ленточка, которая есть у Ильдара.

Третья строка содержит одно целое число  $q$  ( $1 \leq q \leq 200\,000$ ) — количество отрезков-предпочтений у Ильдара.

Каждая из следующих  $q$  строк содержит два целых числа  $l_i$  и  $r_i$  ( $1 \leq l_i \leq r_i \leq n$ ) — границы  $i$ -го отрезка.

### Формат выходных данных

Выведите  $q$  строк, где  $i$ -я из них содержит результат игры на строке  $[l_i, r_i]$ .

### Пример

стандартный ввод	стандартный вывод
9	1
abcabcdce	-1
7	4
1 6	3
4 7	2
5 9	2
6 9	-1
1 9	
3 6	
4 4	

### Замечание

Рассмотрим первый пример.

Если Ильдар выберет отрезок  $[1, 6]$ , то он вырежет строку  $t = abcabc$ . Если разрезать  $t$  на два кусочка  $abc$  и  $abc$ , то строка  $abc$  повторится два раза, а количество различных строк из разрезания будет равно 1. Поэтому результат этой игры 1.

Если Ильдар выберет отрезок  $[4, 7]$ , то он вырежет строку  $t = abcd$ . Эту строку невозможно разрезать на строки так, что будет хотя бы строка, повторяющаяся хотя бы два раза. Поэтому результат этой игры  $-1$ .

Если Ильдар выберет отрезок  $[3, 6]$ , то он вырежет строку  $t = cabc$ . Если разрезать  $t$  на три кусочка  $c$ ,  $ab$  и  $c$ , то строка  $c$  повторится два раза, а количество различных строк будет равно 2. Поэтому результат этой игры 2.

## Задача N. Общие подстроки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дано  $K$  не обязательно различных строк из маленьких латинских букв, с суммарной длиной  $N$ .  $L_i$  определяется как максимальная длина строки, которая встречается как подстрока хотя бы у  $i$  строк из начального набора. Требуется для каждого  $2 \leq i \leq K$  посчитать  $L_i$ .

### Формат входных данных

В первой строке входных данных дано одно число  $K$  ( $1 \leq K \leq 200\,000$ ) — число строк.

В следующих  $K$  строках даны сами строки из начального набора, по одной в строке. Гарантируется, что  $N$  — суммарная длина всех строк не превышает 200 000.

### Формат выходных данных

В  $K - 1$  строке выведите по одному числу —  $L_2, L_3, \dots, L_K$ .

### Пример

стандартный ввод	стандартный вывод
6	5
matter	3
animate	2
pattern	2
thermal	1
domain	
teammate	

## Задача I. Циклический сдвиг

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

Дана строка  $s$ , состоящая из маленьких латинских букв.

Назовем строку  $t = t_1t_2 \dots t_m$  ( $m > 0$ ) хорошей относительно строки  $s$ , если строка  $t$  и ее левый циклический сдвиг  $t' = t_2 \dots t_mt_1$  являются подстроками строки  $s$ .

Вам необходимо найти количество различных хороших строк  $t$  относительно заданной строки  $s$ .

### Формат входных данных

В единственной строке входных данных записана строка  $s$ , состоящая из  $n$  ( $1 \leq n \leq 300\,000$ ) маленьких латинских букв.

### Формат выходных данных

Выведите единственное число — количество хороших строк  $t$  относительно заданной строки  $s$ .

### Примеры

стандартный ввод	стандартный вывод
abaac	7
aaa	3

### Замечание

В первом примере хорошими строками являются следующие: a, b, c, aa, ab, ba, aba.

Во втором примере хорошими являются только три строки: a, aa, aaa.

## Задача J. Рефрен HARD

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

Рассмотрим последовательность  $n$  целых чисел от 1 до  $m$ . Подпоследовательность подряд идущих чисел называется рефреном, если произведение ее длины на количество вхождений в последовательность максимально.

По заданной последовательности требуется найти ее рефрен.

### Формат входных данных

Первая строка входного файла содержит два целых числа:  $n$  и  $m$  ( $1 \leq n \leq 750\,000$ ,  $1 \leq m \leq 10$ ).

Вторая строка содержит  $n$  целых чисел от 1 до  $m$ .

### Формат выходных данных

Первая строка выходного файла должна содержать произведение длины рефрена на количество ее вхождений. Вторая строка должна содержать длину рефрена. Третья строка должна содержать последовательность которая является рефреном.

### Пример

стандартный ввод	стандартный вывод
9 3	9
1 2 1 2 1 3 1 2 1	3
	1 2 1

## Задача К. Общие префиксы-суффиксы

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

Вам даны  $N$  строк  $S_1 \cdots S_N$ . По ним строится новый набор строк  $T_1 \cdots T_M$ , состоящий из всех суффиксов строк  $S_1 \cdots S_N$  (таким образом,  $M = \sum_{i=1}^N |S_i|$ ).

Определим для двух строк  $x$  и  $y$  величину  $\text{lcp}(x, y)$  как длину наибольшего общего префикса строк  $x$  и  $y$ .

Найдите величину:

$$\sum_{i=1}^M \sum_{j=i+1}^M \text{lcp}(T_i, T_j)$$

### Формат входных данных

В первой строке входных данных вводится одно целое число  $N$  ( $1 \leq N \leq 3 \cdot 10^5$ ).

В следующих  $N$  строках вводятся  $S_i$ .  $S_i$  состоит из строчных латинских букв. Гарантируется, что  $\sum_{i=1}^N |S_i| \leq 3 \cdot 10^5$ .

### Формат выходных данных

В единственной строке выходных данных выведите одно число — ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
3 abc ac ab	6
11 ab bb aaa bba baba babb aaaba aabbb a a b	270

## Задача L. Вставить текст

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	6 секунд
Ограничение по памяти:	1024 мегабайта

Алиса и Аня работают копирайтерами. Недавно им пришёл заказ: нужно написать  $k$  текстов (строк) на схожую тематику. Девочки сразу приступили к работе и быстро получили  $k$  строк  $S_1, \dots, S_k$ , каждая из которых имеет длину не более  $m$  и состоит из строчных латинских букв, при этом длина  $S_1$  оказалась в точности равна  $m$ .

Девочки уже давно работают вместе, и у них есть простой способ проверить оригинальность своей работы. Оригинальность оценивается с помощью подстрок полученных текстов. Алиса и Аня считают некоторую строку неоригинальной, если и она, и строка, получающаяся из данной разворотом, встречаются в некоторых из их текстов в качестве подстрок на одних и тех же позициях.

Чтобы ускорить проверку, они разбивают строки на блоки. Разбиение на  $t$  блоков задается последовательностью  $a_0, a_1, \dots, a_t$ , где  $a_0 = 0$ ,  $a_t = m$  и  $a_{i-1} < a_i$  для любого  $1 \leq i \leq t$ . Тогда  $i$ -м блоком называется отрезок целых чисел  $[a_{i-1} + 1; a_i]$ . Блок  $[a_{i-1} + 1; a_i]$  называется интересным, если существует неоригинальная строка, встречающаяся в текстах ровно на позициях, задаваемых данным блоком. Иными словами, блок — интересный, если для каких-то строк  $S_l$  и  $S_r$  (возможно,  $l = r$ ) верно, что  $|S_l|, |S_r| \geq a_i$ , и строка  $S_{l, a_{i-1}+1} S_{l, a_{i-1}+2} \dots S_{l, a_i}$  совпадает со строкой  $S_{r, a_i} S_{r, a_i-1} \dots S_{r, a_{i-1}+1}$ , где  $S_{t,j}$  —  $j$ -й слева символ строки  $S_t$ .

Например, для текстов [abba, ba] последовательности  $(0, 1, 4)$  и  $(0, 1, 2, 3, 4)$  задают корректные разбиения, а последовательности  $(1, 2, 3)$  и  $(0, 1, 1, 4)$  — нет. При этом для разбиения  $(0, 2, 4)$  первый блок  $[1; 2]$  является интересным, поскольку  $S_{1,1} S_{1,2} = S_{2,2} S_{2,1} = ab$ , а второй блок  $[3; 4]$  — нет, поскольку для единственной возможной пары номеров  $l = r = 1$  строки  $S_{1,3} S_{1,4} = ba$  и  $S_{1,4} S_{1,3} = ab$  не совпадают.

Разбиение называется интересным, если каждый блок в этом разбиении интересный. Алиса и Аня хотят найти интересное разбиение текстов на минимальное число блоков, чтобы измерить оригинальность работы. Девочки постарались, чтобы условие этой задачи было современным и прошло тест Бекдел, поэтому теперь помогите им и напишите для них программу, измеряющую оригинальность!

Заметим, что искомая величина корректно определена, так как, разбив строки на  $m$  блоков длины 1, мы получим интересное разбиение (в каждом блоке будет достаточно взять  $l = r = 1$ ).

### Формат входных данных

Во первой строке вводятся три целых числа  $t$ ,  $k$  и  $m$  ( $1 \leq k \leq 200\,000$ ,  $1 \leq m \leq 500\,000$ ) — номер группы, к которой относится данный тест, общее число текстов и длина первого текста.

В  $i$ -й из следующих  $k$  строк вводится  $S_i$  —  $i$ -й текст, состоящий из строчных букв латинского алфавита.

Гарантируется, что  $|S_1| = m$ ,  $|S_i| \leq m$  для любого  $i > 1$ , и суммарная длина всех строк не превосходит 500 000.

### Формат выходных данных

Во первой строке выведите единственное целое число  $t$  — минимальное число блоков в интересном разбиении.

Во второй строке через пробел выведите возрастающую последовательность из  $t - 1$  целого числа  $a_1, \dots, a_{t-1}$  — номеров правых границ всех блоков, кроме последнего (само разбиение имеет вид  $[1; a_1], [a_1 + 1; a_2], \dots, [a_{t-1} + 1; m]$ ).

## Примеры

стандартный ввод	стандартный вывод
0 2 6 abcded cba	2 3
0 6 7 poggers sus amogus tokyo ghoul sodluv	4 3 5 6

## Замечание

В первом примере вторая строка **cba** при перевороте совпадает с первыми тремя символами первой строки **abcded**. Оставшиеся же символы **ded** образуют палиндром, т.е. эта строка совпадает с собой же перевёрнутой. Поэтому мы можем разбить строки на два блока  $[1; 3]$  и  $[4; 6]$ . Легко видеть, что на меньшее число блоков разбить нельзя, ведь **abcded** — не палиндром.

Во втором примере в первом блоке  $[1; 3]$  можно выбрать строку-палиндром **sus**, во втором блоке кусочки пятой (**ghoul**) и шестой (**sodluv**) строк, совпадающих друг с другом при перевороте. В третьем и четвертом блоках выбираем по одной букве из любой строки. Можно показать, что на меньшее число блоков разбить строки нельзя.