

## Задача А. Суффиксный массив

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

Постройте суффиксный массив для заданой строки  $s$ . За копиясту **ЧЁРНЫЙ БАН!!!** <3

### Формат входных данных

Единственная строка входного файла содержит строку  $s$ , состоящую из строчных латинских букв ( $1 \leq |s| \leq 400\,000$ ).

### Формат выходных данных

Выведите  $|s|$  различных чисел — номера первых символов суффиксов строки  $s$  так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

### Примеры

<code>array.in</code>	<code>array.out</code>
ababb	1 3 5 2 4
abacaba	7 5 1 3 6 2 4

## Задача В. LCP для суффиксного массива

Имя входного файла: `sufflcp.in`  
Имя выходного файла: `sufflcp.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 64 мегабайта

Дана строка длины  $N$  и отсортированный массив суффиксов этой строки (т.е. суффиксный массив), вам нужно вычислить LCP. При сортировке строка `a` считается меньше строки `aa`. LCP — наибольший общий префикс двух последовательных суффиксов в суффиксном массиве.

За копирасту опять же **ЧЁРНЫЙ БАН!!!** <3

### Формат входных данных

В первой строке число  $N$  ( $1 \leq N \leq 10^5$ ). На второй строке файла дана  $N$  строчных латинских букв. В третьей строке  $N$  чисел от 1 до  $N$  — суффиксный массив (числом  $i$  кодируется суффикс, начинающийся с  $i$ -го символа).

### Формат выходных данных

Выведите  $N - 1$  число — значения LCP.

### Примеры

<code>sufflcp.in</code>	<code>sufflcp.out</code>
5	1
сасао	0
2 4 1 3 5	2
	0

### Замечание

Суффиксный массив для строки `сасао`:

```
асао
ао
сасао
сао
о
```

## Задача С. Контрольное списывание

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

Сегодня на уроке преподаватель Массивов Автомат Укконеви́ч рассказывал своим ученикам про строки, суффиксные структуры и всё такое. Например, он рассказал им, как сравнить две строки  $A$  и  $B$  лексикографически. Если одна из них является префиксом другой, то более короткая будет лексикографически меньше, иначе необходимо сравнить символы стоящие на первой позиции, в которой они отличаются. Строка с меньшим по номеру в алфавите символом на данной позиции и будет лексикографически меньше.

Чтобы проверить понимание учениками нового материала, Автомат Укконеви́ч дал им следующее задание: найти  $k$ -ю лексикографически непустую уникальную подстроку строки  $S$ .

Так как учитель знает, что Михаил В. и Роман Б. очень любят списывать у известного в узких кругах Максима И., каждый школьник получил своё число  $k$  и вынужден был обратиться к вам за помощью.

### Формат входных данных

В первой строке входного файла находится строка  $S$  ( $|S| \leq 10^5$ ). Вторая строка содержит число  $k$  ( $1 \leq k \leq 10^{18}$ ) — порядковый номер запрашиваемой подстроки.

### Формат выходных данных

Если ответ существует, выведите искомую подстроку строки  $S$ . В противном случае выведите её лексикографически максимальную подстроку.

### Примеры

kthsubstr.in	kthsubstr.out
abacaba 10	acab
abracadabra 10000000000000000000	racadabra

## Задача D. Рефрен

Имя входного файла: `refrain.in`  
Имя выходного файла: `refrain.out`  
Ограничение по времени: 2 секунды  
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  $m$  ( $1 \leq n \leq 150\,000$ ,  $1 \leq m \leq 10$ ).

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.

### Примеры

<code>refrain.in</code>	<code>refrain.out</code>
9 3	9
1 2 1 2 1 3 1 2 1	9
	1 2 1 2 1 3 1 2 1

## Задача Е. Поиск подподстроки в подстроке

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4.5 секунд
Ограничение по памяти:	256 мегабайт

Опытным участникам соревнований по спортивному программированию хорошо известна классическая задача о нахождении количества вхождений подстроки в строку. Обычно она формулируется так: дана строка-образец  $s$  и строка  $t$ , требуется найти количество индексов, начиная с которых строка  $s$  содержится в строке  $t$ .

К сожалению, для решения этой задачи уже придумано множество алгоритмов, поэтому сама по себе она может быть интересна только в качестве упражнения, но не олимпиадной задачи. Однако, как это часто бывает со стандартными задачами, её легко усложнить — представим, что нас интересуют не сами строки  $s$  и  $t$ , а некоторые их подстроки  $s[l_1 \dots r_1]$  и  $t[l_2 \dots r_2]$ .

Как вы уже, наверное, догадались, вам дано  $q$  запросов,  $i$ -й из которых задаёт некоторые подстроки  $\bar{s} = s[l_{1i} \dots r_{1i}]$  и  $\bar{t} = t[l_{2i} \dots r_{2i}]$ . Для каждого такого запроса необходимо посчитать количество вхождений строки  $\bar{s}$  в строку  $\bar{t}$ .

За корнячку **ЧЁРНЫЙ БАН** <3

### Формат входных данных

Первая и вторая строки входных данных содержат строки  $s$  и  $t$  ( $1 \leq |s|, |t| \leq 2 \cdot 10^5$ ) соответственно. Строки состоят из маленьких букв английского алфавита.

В третьей строке задано одно число  $q$  ( $1 \leq q \leq 5 \cdot 10^5$ ) — количество запросов.

Каждая из следующих  $q$  строк содержит по четыре числа  $l_1$ ,  $r_1$ ,  $l_2$  и  $r_2$  ( $1 \leq l_1 \leq r_1 \leq |s|, 1 \leq l_2 \leq r_2 \leq |t|$ ), описывающих очередной запрос.

### Формат выходных данных

Выведите  $q$  чисел — ответы на запросы.

### Примеры

стандартный ввод	стандартный вывод
abb	3
ababababb	1
5	2
1 2 1 7	1
2 3 2 9	4
3 3 4 7	
1 2 2 4	
1 1 1 9	

## Задача F. Болотный доктор

Имя входного файла: `emails.in`  
Имя выходного файла: `emails.out`  
Ограничение по времени: 4 секунды  
Ограничение по памяти: 256 мегабайт

Во время последнего похода к центру Зоны сталкер Штырь угодил в неизвестную ранее аномалию. Дело плохо — у Штыря начали отрастать клыки и покраснели глаза. Теперь от превращения в кровососа Штыря сможет спасти только болотный доктор. . .

Штырь не знает адреса электронной почты доктора, но предполагает, что он есть в КПК погибшего сталкера Семяцкого, который Штырь недавно нашёл. Беда в том, что хитрый Семяцкий не использовал адресную книгу, а прятал адреса электронной почты в содержимом большого текстового файла. Несмотря на то, что Штырь может прочитать весь этот файл, тайну того, какой фрагмент файла является адресом доктора, Семяцкий унёс с собой в могилу.

Штырь решил перебрать все фрагменты текста, которые могут являться адресами, отправляя по одному письму на каждый из них. Сколько же писем ему придётся отправить?

Адрес электронной почты в окрестностях Зоны состоит из имени пользователя и домена, разделённых символом «@». Имя пользователя и домен — непустые строки, состоящие из строчных английских букв и точек. При этом они не могут начинаться на точку, заканчиваться точкой и содержать две точки подряд.

### Формат входных данных

Во входных данных записано содержимое файла на КПК Семяцкого. В файле могут встречаться только символы с ASCII-кодами от 32 до 126 и переводы строк. Объём входных данных не превосходит  $10^6$  байт.

### Формат выходных данных

Выведите, сколько различных электронных адресов содержит файл Семяцкого.

### Примеры

<code>emails.in</code>	<code>emails.out</code>
<code>a.b@c.d</code> <code>b@c.de.</code> <code>@qq</code> <code>q@.q</code>	5

### Замечание

В файле встречаются следующие адреса: «`a.b@c`», «`a.b@c.d`», «`b@c`», «`b@c.d`», «`b@c.de`».

## Задача G. Частотность

Имя входного файла: `frequent.in`  
Имя выходного файла: `frequent.out`  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

В данный момент астробиологи ведут работу по изучению форм жизни на планете Альфабет. Жизнь там основана на ДНК, составленных из 26 различных нуклеотидов. Таким образом, ДНК каждой формы жизни на планете Альфабет может быть представлена строкой, состоящей из строчных букв английского алфавита. Астробиологи уже получили последовательности ДНК для  $K$  форм жизни, суммарная длина этих  $K$  последовательностей равняется  $N$ . Возможна ситуация, что ДНК некоторых форм жизни совпадают.

Теперь они хотели бы выделить некоторые нити (подстроки) данных ДНК, которые встречаются у различных форм жизни. Обозначим через  $L(i)$  (здесь  $2 \leq i \leq K$ ) максимальную длину нити (подстроки), состоящей из последовательных нуклеотидов, которая встречается хотя бы у  $i$  форм жизни. Обратите внимание, что  $L(i)$  может быть равно 0.

Вычислите значения функции  $L(i)$  для всех  $i$  от 2 до  $K$ .

### Формат входных данных

В первой строке записано целое число  $K$ , означающее количество форм жизни, для которых была выделена последовательность ДНК. В каждой из последующих  $K$  строк записана непустая строка, состоящая из строчных букв английского алфавита. ( $2 \leq N \leq 200\,000$ ,  $2 \leq K \leq N$ ).

### Формат выходных данных

Выходной файл должен содержать  $K - 1$  строку со значениями  $L(2), L(3), \dots, L(K)$ , каждое на отдельной строке.

### Примеры

<code>frequent.in</code>	<code>frequent.out</code>
6	5
matter	3
animate	2
pattern	2
thermal	1
domain	
teammate	

## Задача Н. Юнг и ковёр

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

Элоди Юнг — талантливая актриса; вы могли видеть её в таких фильмах как “Боги Египта”, “.I. Джо: Бросок кобры ” или “Девушка с татуировкой дракона”. Недавно Элоди купила восхитительный коврик, состоящий из  $n$  спитых подряд кусочков ткани, каждый из которых покрашен в какой-то из 26 цветов. Для простоты будем считать этот коврик диаграммой Ахо длины  $n$  (каждому цвету соответствует один из символов латинского алфавита).

Элоди хочет вырезать из этого длинного коврика маленький коврик для своей мамы (мама Элоди — француженка и ценит уютные предметы декора). Для этого она выберет какой-то непрерывный непустой подотрезок коврика, вырежет его — это и будет подарок для её мамы.

Но это еще не все. Из оставшихся частей она хочет сшить точно такой же коврик. Для этого она собирается вырезать из левой и правой оставшихся частей по кусочку ткани и сшить их вместе в таком же порядке — это и будет коврик для неё. При этом Элоди хочет, чтобы эти коврики были **абсолютно** одинаковыми — тем самым она покажет маме свою любовь к ней. Ей стало интересно, сколькими способами можно вырезать коврик.

Формально говоря, она хочет выяснить, сколько существует таких пар индексов  $i, j$ , ( $1 \leq i \leq j \leq n$ ), для которых найдутся такие  $i_1, j_1, i_2, j_2$  ( $1 \leq i_1 \leq j_1 < i, j < i_2 \leq j_2 \leq n$ ), что  $s[i..j] = s[i_1..j_1] + s[i_2..j_2]$ . ( $s[l..r]$  — это поддиаграммы диаграммы  $s$ , с  $l$ -го символа по  $r$ -й символ включительно).

Сама Элоди не в состоянии посчитать это число, поэтому она попросил вас помочь ей.

### Формат входных данных

В первой и единственной строке входного файла дана диаграмма Ахо  $s$  ( $1 \leq |s| \leq 100\,000$ ).

### Формат выходных данных

В первой и единственной строке выходного файла выведите одно число — количество способов вырезать коврик.

### Примеры

стандартный ввод	стандартный вывод
aaaa	1
abababb	3

### Замечание

В первом примере существует единственная подходящая пара индексов (2, 3). Во втором примере существуют три подходящие пары индексов (3, 4), (5, 6) и (4, 6).

## Задача I. Путешествие по строке

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	7.5 секунд
Ограничение по памяти:	512 мегабайт

Скажем, что последовательность строк  $t_1, \dots, t_k$  является *путешествием* длины  $k$ , если для всех  $i > 1$   $t_i$  является подстрокой  $t_{i-1}$  строго меньшей длины. Например,  $\{ab, b\}$  является путешествием, а  $\{ab, c\}$  или  $\{a, a\}$  — нет.

Определим *путешествие по строке*  $s$  как путешествие  $t_1, \dots, t_k$ , все строки которого могут быть вложены в  $s$  так, чтобы существовали (возможно, пустые) строки  $u_1, \dots, u_{k+1}$ , такие, что  $s = u_1 t_1 u_2 t_2 \dots u_k t_k u_{k+1}$ . К примеру,  $\{ab, b\}$  является *путешествием по строке* для  $abb$ , но не для  $bab$ , так как соответствующие подстроки расположены справа налево.

Назовём *длиной путешествия* количество строк, из которых оно состоит. Определите максимально возможную длину путешествия по заданной строке  $s$ .

### Формат входных данных

В первой строке задано целое число  $n$  ( $1 \leq n \leq 500\,000$ ) — длина строки  $s$ .

Во второй строке содержится строка  $s$ , состоящая из  $n$  строчных английских букв.

### Формат выходных данных

Выведите одно число — наибольшую длину путешествия по строке  $s$ .

### Примеры

стандартный ввод	стандартный вывод
7 abcdbcc	3
4 bbcb	2

### Замечание

В первом примере путешествием по строке наибольшей длины является  $\{abcd, bc, c\}$ .

Во втором примере подходящим вариантом будет  $\{bb, b\}$ .

## Задача J. Общие префиксы

Имя входного файла: стандартный ввод  
 Имя выходного файла: стандартный вывод  
 Ограничение по времени: 4 секунды  
 Ограничение по памяти: 1024 мегабайта

Обозначим  $LCP(X, Y)$  за длину наибольшего общего префикса строк  $X$  и  $Y$ . Например  $LCP("aboba" "abaca") = 2$ , а  $LCP("aaa" "aaaa") = 3$ .

Дана строка  $S$  длины  $N$ . Обозначим также суффикс строки  $S$ , начинающийся с  $i$ -го символа за  $S_i$ . Для всех  $k = 1, \dots, n$  Найдите сумму  $LCP(S_k, S_1) + LCP(S_k, S_2) + \dots + LCP(S_k, S_n)$

### Формат входных данных

В первой строке вводится число  $N$  ( $1 \leq N \leq 10^6$ )

Во второй записана строка  $S$ , состоящая из малых букв английского алфавита

### Формат выходных данных

Выведите  $N$  чисел, по одной в каждой строке — значение суммы для  $k = 1, \dots, n$

### Примеры

стандартный ввод	стандартный вывод
3 abb	3 3 2
11 mississippi	11 16 14 12 13 11 9 7 4 3 4