

Задача А. 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 искомых 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.

Примеры

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

Задача В. Суффиксный массив

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

Данна строка, требуется построить суффиксный массив для этой строки. Суффиксный массив — лексикографически отсортированный массив всех суффиксов строки. Каждый суффикс задается целым числом — позицией начала.

Строка s лексикографически меньше строки t , если есть такое i , что $s_i < t_i$ и $s_j = t_j$ для всех $j < i$. Или, если такого i не существует и строка s короче строки t .

Здесь s_i — код i -го символа строки s .

Формат входных данных

Файл состоит из единственной строки. Эта строка — **английский литературный текст**. Длина текста не превосходит 10^5 . Коды всех символов в тексте от 32 до 127.

Формат выходных данных

Выведите N чисел — суффиксный массив данной строки.

Примеры

suffarray.in	suffarray.out
99 bottles of beer.	14 3 11 19 2 1 15 4 16 17 9 13 8 12 5 18 10 7 6

Задача С. 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.

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

sufflcp.in	sufflcp.out
5 сасао 2 4 1 3 5	1 0 2 0

Замечание

Суффиксный массив для строки **сасао**:

асао
ао
сасао
сао
о

Задача D. Бинарные строки

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

Строка называется *бинарной*, если она состоит только из символов '0' и '1'.

Рассмотрим бинарную строку w длины n . *Суффиксным массивом* строки w называется массив $a[1..n]$ такой, что строка $w[a[i]..n]$ является i -ым в лексикографическом порядке суффиксом строки w . Например, в результате сортировки суффиксов строки $w = \text{"001011"}$ они будут расположены следующим образом: "001011", "01011", "011", "1", "1011", "11". Следовательно, суффиксный массив для строки w выглядит так: (1, 2, 4, 6, 3, 5).

Вам дан суффиксный массив a неизвестной строки w . Требуется восстановить строку w .

Формат входных данных

Первая строка входного файла содержит n — длину строки w ($1 \leq n \leq 300\,000$). Вторая строка содержит n различных целых чисел в диапазоне от 1 до n — суффиксный массив строки w .

Формат выходных данных

Выведите единственную строку — искомую бинарную строку w , суффиксный массив которой совпадает с массивом, заданным во входных данных. Если таких строк несколько, выведите любую из них. В случае, если таких строк не существует, выведите **"Error"**.

Примеры

binary.in	binary.out
6 1 2 4 6 3 5	001011

Задача Е. 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 Укконевич рассказывал своим ученикам про строки и все такое. Например, он рассказал им, что подстрока A лексикографически меньше подстроки B , если она является ее префиксом, или если символ, стоящий в строке A в первой позиции, в которой эти строки различаются, имеет меньший номер в алфавите, чем аналогичный символ в строке B .

Чтобы проверить понимание учениками нового материала, Автомат Укконевич выдал им следующее задание: найти k -ую лексикографически непустую уникальную подстроку строки S .

Так как учитель знает, что Михаил Вишневский и Роман Бабенко очень любят списывать у известного в узких кругах Максима Иванова, каждый школьник получил свое 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 ее лексикографически максимальную подстроку.

Примеры

<code>kthsubstr.in</code>	<code>kthsubstr.out</code>
abacaba 10	acab
abracadabra 10000000000000000000	racadabra

Задача F. 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 1 2 1 2 1 3 1 2 1	9 3 1 2 1