

Задача А. Вася и Циклические Сдвиги

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

Однажды Василий проголодался и пошел на кухню подкрепиться. Но каково было его удивление, когда, открыв холодильник, он обнаружил там не привычные продукты, а строку! Причем не просто строку, а заикленную, прямо как бублик (ассоциации с едой часто приходят Васе в голову, когда он голоден). Загадочная строка заинтересовала любопытного, но все еще голодного Василия, и он начал ее вертеть в руках. Например, если вертеть строку `abacaba`, то можно получить следующие строки:

- `abacaba`
- `bacabaa`
- `acabaab`
- `cabaaba`
- `abaabac`
- `baabaca`
- `aabacab`

И тут Василия осенило: если он сможет посчитать, сколько раз в процессе кручения строки получается лексикографически минимальная строка, то еда магическим образом появится в холодильнике (странные идеи часто приходят Васе в голову, когда он голоден). Помогите Васе, иначе он так и будет сидеть голодным.

Более формально, вам дана строка. *Циклическим сдвигом* строки s длины n называется строка, полученная из исходной путем отбрасывания первых $0 \leq k < n$ символов и приписывания их в конец. Необходимо посчитать, сколько раз среди всех циклических сдвигов строки встречается лексикографически минимальный циклический сдвиг.

Формат входных данных

Единственная строка входного файла содержит строку S , найденную Василием. Она непуста, состоит из маленьких латинских букв, и её длина не превосходит 10 000 000.

Формат выходных данных

Выведите единственное число — искомое количество минимальных циклических сдвигов.

Примеры

стандартный ввод	стандартный вывод
<code>aaaa</code>	4
<code>abacaba</code>	1

Замечание

В первом примере минимальным циклическим сдвигом является строка `aaaa`. Во втором примере минимальным циклическим сдвигом является строка `aabacab`.

Задача В. Задачка на строчечки

Имя входного файла: inputik.txt
Имя выходного файла: outputik.txt
Ограничение по времени: 2 секундочки
Ограничение по памяти: 256 мегабайтиков

В стандартном поточике вводика или файлик
е
кой. На следующей строчечке программочк
ь а і
н к; эти словечки мы назовём словарик п
е о о н р
н в utputik.txt N строчечек. В і-ой ма и
д о о . й т
о к - количюсик (сколько штучеч с д і
х у к е т В ё k
с б и о ем через пробельчик для к р а т .
и л к т) о ш т
х й л а в всех вхожденъечек к ч а ч х
м и а е з о а в е и t
ё к ф с к ртированном про э ж х ч п с
в с и , и о я т д о к р е в
о н в ч й л с рочечек нач д о о же о л а
з и о а т т и о й г д г к ш
а т и е к ч о с ничек. н ч о е п р о а
на л о ь а и а к с н р а
л и в н н в х де с ястю е т в ь о м N п
ы р е е . р х е г м , р
м х а е н и ю св и к и с к е д н и о о ч р о о
и к п д к у ч ж е а ч а г
ю к и о и к ь н е н д о x с и в и к ч e д к м к р
у ь д : х с е м а в а
р н о к с к e d н и e т и д e в ы в а к ч и ь н с o м
о e в o и т ч д с м
т л ы л в а к и р a в o л с з и і и к ч e ч o p k o л o
о а в e а л e ч
к м с и ч o к ь л o x с e n и т с e в ы в a н ж л o d ж д к
к н у а
, e e ч и ч o т o п й н т р a d н a т c a н и т s e v ы в a ю
к ж щ н
о х e т з и y к ч e v o l s o п - х a k ч e ч o p t c N х и а
в й
о к у б х и k c н и т a л х и k ь н e л a m з и y k ч e ч o p t c т ё д

Формат входных данных

В стандартном поточике вводика или файлике `inputik.txt` ваша программочка найдёт строчечку из маленьких латинских буковок, которую мы назовём исходненькой. На следующей строчечке программочка найдёт числище N ($1 \leq N \leq 1\,000\,000$), а в следующих N строчечках — по словечку из тех же маленьких латинских буковок; эти словечки мы назовём словариком. Суммарненькая суммочка длинниц словечек из словарика не превосходит $1\,000\,000$.

Формат выходных данных

Ваша программочка должна вывести на стандартный поточичек выводика или в файл `outputik.txt` N строчечек. В i -ой строчечке программочка должна вывести несколько чиселок: первое чиселко — количюсик (сколько штукечек) вхожденьчек строчечки i из словарика в исходненькой, затем через пробельчик для каждого вхожденьчика выведите индексики началиков всех вхожденьчик этой строчечки в исходненькую в отсортированном порядочке. Индексики всех строчечек начинаются с единичек. Няшечки-преподавашечки гарантируют, что колчюсик вхожденьчик не превосходит $1\,000\,000$.

Пример

<code>inputik.txt</code>	<code>outputik.txt</code>
<code>abrachkacadabrachka</code>	<code>2 1 12</code>
<code>4</code>	<code>1 9</code>
<code>abrachka</code>	<code>2 1 12</code>
<code>cadabrachka</code>	<code>0</code>
<code>ab</code>	
<code>marazmik</code>	

Задача С. Анаграммные подстроки

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

У Андрюши есть маленькая сестричка Аня. Она любит писать сообщения своему другу Гоше. Она хочет, чтобы никто не мог прочитать ее сообщения, поэтому она шифрует их подстановочным шифром. Подстановочный шифр заменяет каждый символ в сообщении на какой-либо еще, при этом равные символы заменяются на равные, а различные — на различные. Например, при шифровании с помощью подстановочного шифра $e \rightarrow a, l \rightarrow b, o \rightarrow w, v \rightarrow c$ слово «love» оказывается зашифровано как «bwca».

Андрюша недавно перехватил одно из Аниных сообщений t и хочет выяснить, встречается ли там текст s . А именно, он хочет найти все позиции i , такие что существует подстановочный шифр, такой что $t_i \dots t_{i+|s|-1}$ представляет собой зашифрованную версию s . Будем называть такие позиции потенциальными вхождениями s в t . Помогите Андрюше найти все потенциальные вхождения s в t .

Формат входных данных

Первая строка входного файла содержит t . Вторая строка входного файла содержит s . Каждая строка состоит из символов с ASCII кодами от 33 до 126. Длина s не превышает длины t . Длина t не превышает 200 000. Обе строки непусты.

Формат выходных данных

Первая строка выходного файла должна содержать k — количество потенциальных вхождений s в t . Вторая строка должна содержать k целых чисел — позиции потенциальных вхождений. Позиции в строке нумеруются, начиная с 1. Позиции следует перечислить в возрастающем порядке.

Примеры

стандартный ввод	стандартный вывод
abacabadabacaba aba	7 1 3 5 7 9 11 13
abacabadabacaba love	0

Задача D. Подстрока

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

Недавно разведка перехватила зашифрованное сообщение — строку s . Все ресурсы аналитического центра, в котором вы работаете, были брошены на его декодирование. Ваш отдел занимается шифрами нового поколения. На данный момент известно всего n таких шифров. Для каждого из них есть три характерных параметра — целые числа l, r и строка t . Пусть строка g была получена в результате применения этого метода. Тогда строка $g_l g_{l+1} \dots g_{r-1} g_r$ (здесь g_i — это i -й символ строки g) содержит t как подстроку.

Вам поручено определить для каждого типа шифрования, могло ли сообщение s быть получено в результате его применения.

Формат входных данных

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 100\,000$, где $|s|$ — длина строки s).

Вторая строка входного файла содержит целое число n — количество типов шифрования ($1 \leq n \leq 100\,000$). Последующие n строк содержат по два целых числа l_i, r_i и строку t_i , разделенные пробелами — характерные параметры i -го метода шифрования ($1 \leq l_i \leq r_i \leq |s|$).

Все строки состоят из строчных букв латинского алфавита. Суммарная длина всех t_i не превосходит 100 000.

Формат выходных данных

Выведите одну строку — для каждого типа шифрования «+», если сообщение s могло быть получено в результате его применения, или «-» в противном случае.

Пример

стандартный ввод	стандартный вывод
frommarsiam 3 6 10 i 2 11 am 1 9 human	++-

Задача Е. Чтение строк

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

Дано дерево T состоящее из n вершин (пронумерованных целыми числами от 1 до n). В каждой вершине записана некоторая буква. Корень дерева расположен в вершине 1.

Рассмотрим поддерево T_v некоторой вершины v . Вдоль любого простого пути, начинающегося в v и заканчивающегося в некоторой вершине $u \in T_v$ (возможно, в самой v), можно прочитать некоторую строку. Обозначим количество **различных** строк, которые можно прочитать таким способом как $\text{dif}(v)$.

Дополнительно: для каждой вершины v дано целое число c_v . Нас интересуют вершины, в которых значение $\text{dif}(v) + c_v$ как можно больше.

Вы должны вычислить две величины — максимальное значение $\text{dif}(v) + c_v$ и количество вершин v с максимальным $\text{dif}(v) + c_v$.

Формат входных данных

В первой строке входных данных записано единственное целое число n ($1 \leq n \leq 300\,000$) — количество вершин в дереве T .

Во второй строке записано n целых чисел c_i ($0 \leq c_i \leq 10^9$).

В третьей строке записана строка s , состоящая из n строчных букв английского алфавита, — i -й символ этой строки соответствует букве, записанной в вершине i .

Далее следует $n - 1$ строка с описанием рёбер дерева T . Каждая из них содержит два целых числа u и v ($1 \leq u, v \leq n$), обозначающих ребро, которое соединяет вершины u и v .

Гарантируется, что входные данные описывают дерево.

Формат выходных данных

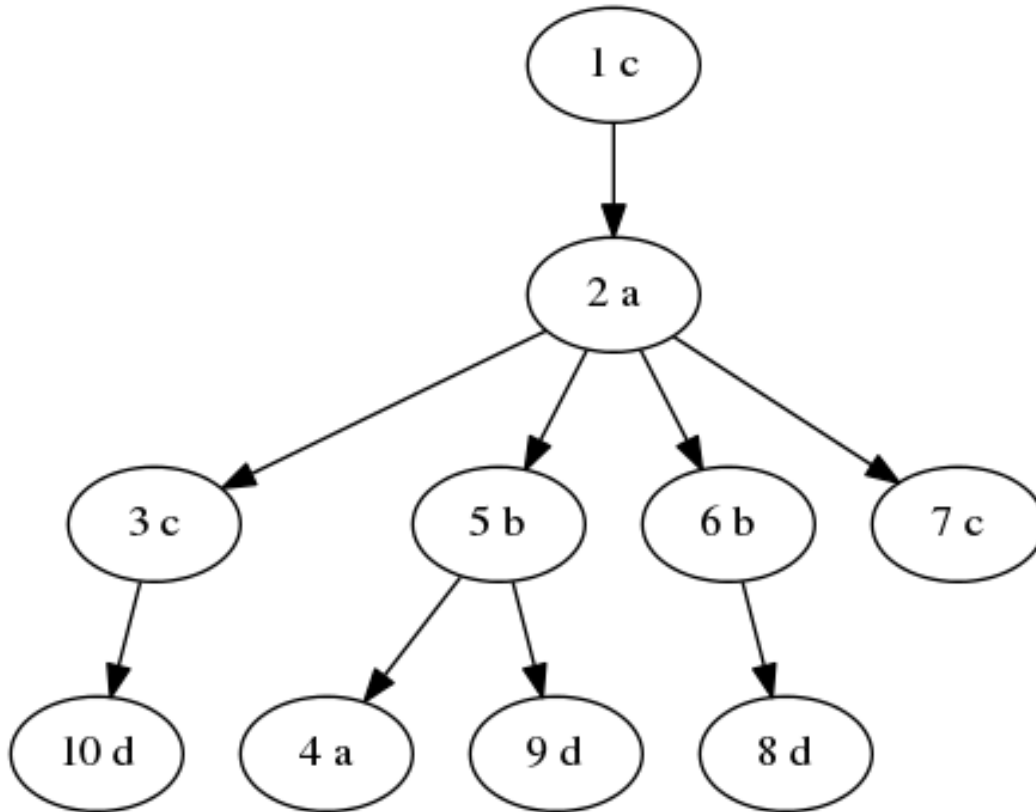
Выведите два числа — значение $m = \max(\text{dif}(i) + c_i)$ для всех $1 \leq i \leq n$ и количество вершин v , для которых $m = \text{dif}(v) + c_v$.

Примеры

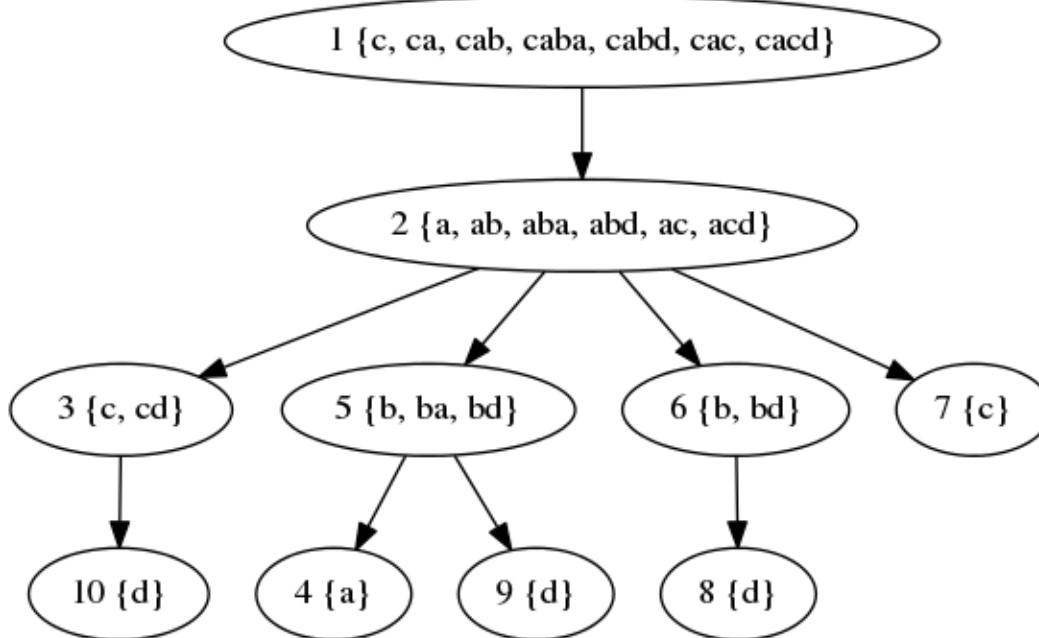
стандартный ввод	стандартный вывод
<pre>10 1 2 7 20 20 30 40 50 50 50 cacabbccddd 1 2 6 8 7 2 6 2 5 4 5 9 3 10 2 5 2 3</pre>	<pre>51 3</pre>
<pre>6 0 2 4 1 1 1 raaaba 1 2 2 3 2 4 2 5 3 6</pre>	<pre>6 2</pre>

Замечание

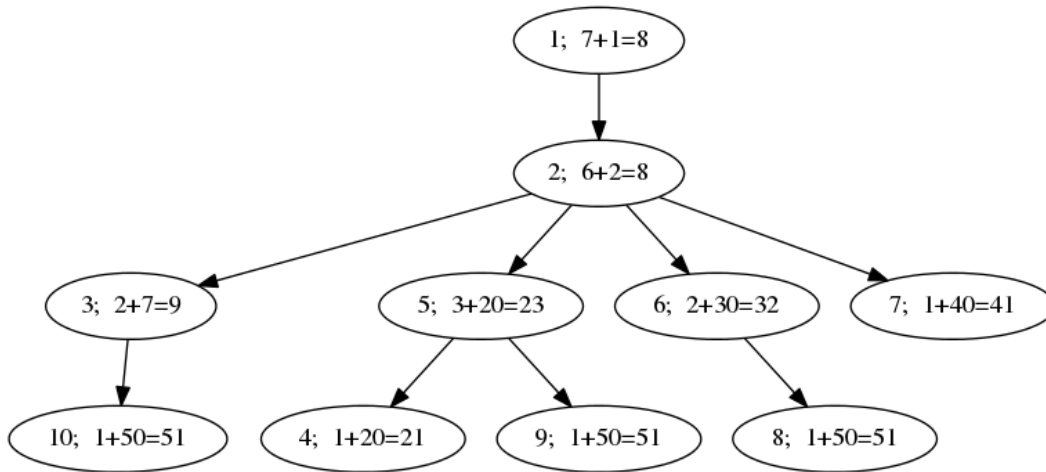
В первом примере дерево выглядит следующим образом:



Наборы строк, которые могут быть прочитаны из вершин:



Наконец, значения $\text{dif}(v) + c_v$ таковы:



Во втором примере значения $\text{dif}(1..n)$ таковы: $(5, 4, 2, 1, 1, 1)$. Различные строки, которые можно прочитать из вершины 2 таковы: a, aa, aaa, ab ; обратите внимание, что aa может быть прочитано как на пути до вершины 3, так и на пути до вершины 4.

Задача F. Вирусы

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

Комитет По Исследованию Бинарных Вирусов обнаружил, что некоторые последовательности единиц и нулей являются кодами вирусов. Комитет изолировал набор кодов вирусов. Последовательность из единиц и нулей называется безопасной, если никакой ее подотрезок (т.е. последовательность из соседних элементов) не является кодом вируса. Сейчас цель комитета состоит в том, чтобы установить, существует ли бесконечная безопасная последовательность из единиц и нулей.

Формат входных данных

Первая строка входного файла содержит одно целое число N , равное количеству всех вирусных кодов. Каждая из следующих n строк содержит непустое слово, составленное из символов 0 и 1 — код вируса. Суммарная длина всех слов не превосходит 30000.

Формат выходных данных

Первая и единственная строка выходного файла должна содержать слово:

- **ТАК** — если бесконечная, безопасная последовательность из нулей и единиц существует;
- **НIE** — в противном случае.

Примеры

стандартный ввод	стандартный вывод
3 01 11 00000	NIE
3 011 11 0000	ТАК

Задача G. Тандемные повторы

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

Дана строка s длины n .

Тандемным повтором в ней называются два вхождения какой-либо подстроки подряд. Иными словами, тандемный повтор описывается парой индексов $i < j$ такими, что подстрока $s[i \dots j]$ — это две одинаковые строки, записанные подряд.

От вас требуется посчитать количество пар индексом $i < j$ таких, что подстрока $s[i \dots j]$ является тандемным повтором.

Формат входных данных

Во входном файле находятся не более 30 тестов. Каждый тест состоит из единственной непустой строки, состоящей из символов **A,C,G,T**. Длина строки не превосходит 10^5 . Входной файл заканчивается строкой **0**.

Формат выходных данных

Для каждого теста выведите единственное число — количество тандемных повторов. Числа разделяйте переводами строк.

Пример

стандартный ввод	стандартный вывод
AGGA	1
AGAG	1
ATTTCGATTTCGATTTCG	9
AAAA	4
0	

Задача Н. Шок контент

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

Вам даны две последовательности длины N : $a = \{a_0, \dots, a_{N-1}\}$ и $b = \{b_0, \dots, b_{N-1}\}$, состоящие из целых положительных чисел.

Ёж выберет целое число k , такое что $0 \leq k < N$ и целое положительное число x . После этого он создаст новую последовательность длины N $a' = \{a'_0, \dots, a'_{N-1}\}$ так, что $a'_i = a_{(i+k) \bmod N} \oplus x$.

Найдите все пары (k, x) такие, что a' будет равняться b .

Формат входных данных

Первая строка входных данных содержит целое число N ($1 \leq N \leq 2 \cdot 10^5$) — количество элементов в последовательностях.

Вторая строка содержит N целых чисел a_i ($0 \leq a_i \leq 2^{30}$) — элементы последовательности a .

Третья строка содержит N целых чисел b_i ($0 \leq b_i \leq 2^{30}$) — элементы последовательности b .

Формат выходных данных

Выведите все пары (k, x) , что a' будет равняться b .

На каждой строке вывода должна содержаться ровно 1 пара.

Пары должны идти в порядке возрастания k (в случае равенства k в порядке возрастания x)

Если подходящих пар не существует, то вывод должен быть пустым.

Примеры

стандартный ввод	стандартный вывод
3 0 2 1 1 2 3	1 3
5 0 0 0 0 0 2 2 2 2 2	0 2 1 2 2 2 3 2 4 2
6 0 1 3 7 6 4 1 5 4 6 2 3	2 2 5 5
2 1 2 0 0	

Задача I. Удаление

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 128 мегабайт

Дано n различных строк s_i . Требуется найти количество таких пар (i, j) , что $i \neq j$, а так же s_j можно получить из s_i каждый раз удаляя только первый или второй символ из s_i .

Например из строки `abcxux` можно получить строку `сux`: `abcxux` \rightarrow `асxux` \rightarrow `сxux` \rightarrow `сux`.

Формат входных данных

Первая строка содержит число n ($2 \leq n \leq 200000$) — количество строк.

i -я из следующих n строк содержит строку s_i ($1 \leq |S|$). Гарантируется, что s_i состоит из строчных букв латинского алфавита.

Так же гарантируется, что $|s_1| + \dots + |s_n| \leq 10^6$.

Примеры

стандартный ввод	стандартный вывод
3 abcxux сux abc	1
6 b a abc c d ab	5