

Задача А. Префикс-функция

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

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её префикс-функцию $\pi[i]$.

Формат входных данных

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Формат выходных данных

Выведите N чисел — значения префикс-функции для каждой позиции, разделённые пробелом.

Примеры

<code>prefix-function.in</code>	<code>prefix-function.out</code>
abracadabra	0 0 0 1 0 1 0 1 2 3 4

Задача В. Z-функция

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

Дана непустая строка S , длина которой N не превышает 10^6 . Будем считать, что элементы строки нумеруются от 1 до N .

Требуется для всех i от 1 до N вычислить её z-функцию $z[i]$.

Формат входных данных

Одна строка длины N , $0 < N \leq 10^6$, состоящая из маленьких латинских букв.

Формат выходных данных

Выведите N чисел — значения z-функции для каждой позиции, разделённые пробелом.

Примеры

<code>z-function.in</code>	<code>z-function.out</code>
abracadabra	11 0 0 1 0 1 0 4 0 0 1

Задача С. К-я строка

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

Реализуйте структуру данных, которая поддерживает следующие операции:

- добавить в словарь строку S ;
- найти в словаре k -ю строку в лексикографическом порядке.

Известно, что изначально словарь пуст.

Формат входных данных

Первая строка входного файла содержит натуральное число N — количество команд ($N \leq 10^5$). Последующие N строк содержат по одной команде каждая.

Команда записывается либо в виде числа k , либо в виде строки S , которая может состоять только из строчных латинских букв. Гарантируется, что при запросе k -й строки она существует. Также гарантируется, что сумма длин всех добавляемых строк не превышает 10^5 .

Формат выходных данных

Для каждого числового запроса k выходной файл должен содержать k -ю в лексикографическом порядке строчку из словаря на момент запроса. Гарантируется, что суммарная длина строк в выходном файле не превышает 10^5 .

Примеры

<code>kthstr.in</code>	<code>kthstr.out</code>
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

Задача D. Мультимножество Василия

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

У автора уже закончились истории про Василия, поэтому он просто написал формальную постановку задачи.

У вас есть q запросов и мультимножество A , изначально содержащее только число 0. Запросы бывают трёх видов:

- «+ x » — добавить в мультимножество A число x .
- «- x » — удалить одно вхождение числа x из мультимножества A . Гарантируется, что хотя бы одно число x в этот момент присутствует в мультимножестве.
- «? x » — вам даётся число x , требуется вычислить $\max_{y \in A} x \oplus y$, то есть максимальное значение побитового исключающего ИЛИ (также известно как XOR) числа x и какого-нибудь числа y из мультимножества A .

Мультимножество — это множество, в котором разрешается несколько одинаковых элементов.

Формат входных данных

В первой строке входных данных содержится число q ($1 \leq q \leq 200\,000$) — количество запросов, которые требуется обработать Василию.

Каждая из последующих q строк входных данных содержит один трёх символов «+», «-» или «?» и число x_i ($1 \leq x_i \leq 10^9$). Гарантируется, что во входных данных встречается хотя бы один запрос «?».

Обратите внимание, что число 0 всегда будет присутствовать в мультимножестве.

Формат выходных данных

На каждый запрос типа «?» выведите единственное целое число — максимальное значение побитового исключающего ИЛИ для числа x_i и какого-либо числа из мультимножества A .

Примеры

стандартный ввод	стандартный вывод
10	11
+ 8	10
+ 9	14
+ 11	13
+ 6	
+ 1	
? 3	
- 8	
? 3	
? 8	
? 11	

Замечание

После первых пяти операций в мультимножестве A содержатся числа 0, 8, 9, 11, 6 и 1.

Ответом на шестой запрос будет число $11 = 3 \oplus 8$ максимальное из чисел $3 \oplus 0 = 3$, $3 \oplus 9 = 10$, $3 \oplus 11 = 8$, $3 \oplus 6 = 5$ и $3 \oplus 1 = 2$.

Задача Е. Период строки

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

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 ($1 \leq |s| \leq 5 \cdot 10^6$). Строка состоит из букв латинского алфавита.

Формат выходных данных

Выведите длину искомой строки t .

Примеры

<code>period.in</code>	<code>period.out</code>
abcabcabc	3

Задача F. Неточное совпадение

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

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

<code>inexact-matching.in</code>	<code>inexact-matching.out</code>
aaaa	4
Caabdbaaaa	1 2 6 7

Задача G. Общая подпоследовательность

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

У Юнга есть массив A из N элементов.

Он хочет найти максимизировать величину $(A[l_1] \oplus A[l_1+1] \oplus \dots \oplus A[r_1]) + (A[l_2] \oplus A[l_2+1] \oplus \dots \oplus A[r_2])$,
где $1 \leq l_1 \leq r_1 < l_2 \leq r_2 \leq N$. Юнг – обычный учёный, помогите ему.

Формат входных данных

На первой строке число N ($2 \leq N \leq 10^6$). На второй строке N целых чисел от 0 до 10^9 .

Формат выходных данных

Одно число – максимум.

Примеры

subseq.in	subseq.out
5 1 2 3 1 2	6

Задача Н. Последнее слово Джека

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

Джек недавно прочитал на заборе интересное и новое для него слово. Оно настолько понравилось Джеку, что он захотел сам придумать ещё какое-нибудь интересное слово. Но только ничего у него не вышло — все придуманные им слова состояли из префиксов исходного слова и поэтому не приносили радости. Он стал придумывать всё более и более длинные слова, но ни одно из них не было оригинальным. . .

И вот настало время Джеку сказать своё последнее слово.

Формат входных данных

Первая строка содержит интересное слово, которое было написано на заборе. Вторая строка содержит последнее слово Джека. Длины слов не превосходят 75 000, слова непустые и состоят из строчных латинских букв.

Формат выходных данных

Если Джек так ничего и не придумал своего, выведите первой строкой `No`. В этом случае покажите Джеку, как разбить его последнее слово на несколько частей, каждая из которых является исходным словом или его непустым префиксом — выведите все эти части во второй строке, разделяя их пробелом. Если же такого разбиения нет, и последнее слово было за Джеком, выведите единственной строкой `Yes`.

Примеры

<code>prefix.in</code>	<code>prefix.out</code>
abracadabra abrabracada	No abr abracada
abracadabra arbadacarba	Yes