

Задача А. Встреченные ранее числа

Имя входного файла: `metbefore.in`
Имя выходного файла: `metbefore.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Формат входных данных

Во входной строке записана последовательность из не более, чем 100 000 чисел через пробел. Каждое из чисел не превышает 10^9 по абсолютному значению.

Формат выходных данных

Для каждого числа выведите слово **YES** (в отдельной строке), если это число ранее встречалось в последовательности или **NO**, если не встречалось.

Примеры

<code>metbefore.in</code>	<code>metbefore.out</code>
1 2 3 2 3 4	NO NO NO YES YES NO

Задача В. Функция

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

Вычислите функцию: $f(n) = \begin{cases} 1 & \text{если } n \leq 2 \\ f(\lfloor 6 * n / 7 \rfloor) + f(\lfloor 2 * n / 3 \rfloor) & \text{если } n \bmod 2 = 1 \\ f(n - 1) + f(n - 3) & \text{если } n \bmod 2 = 0 \end{cases}$

Формат входных данных

Входные данные содержат натуральное число n ($1 \leq n \leq 10^{18}$).

Формат выходных данных

Выведите значение функции по модулю 2^{32} .

Примеры

<code>function.in</code>	<code>function.out</code>
7	10

Замечание

Здесь обязательно реализовать свою хеш-таблицу.

Задача С. Обмен

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

Пусть все натуральные числа исходно организованы в список в естественном порядке. Разрешается выполнить следующую операцию: $swap(a, b)$. Эта операция возвращает в качестве результата расстояние в текущем списке между числами a и b и меняет их местами.

Задана последовательность операций $swap$. Требуется вывести в выходной файл результат всех этих операций.

Формат входных данных

Первая строка входного файла содержит число n ($1 \leq n \leq 200\,000$) — количество операций. Каждая из следующих n строк содержит по два числа в диапазоне от 1 до 10^9 — аргументы операций $swap$.

Формат выходных данных

Для каждой операции во входном файле выведите ее результат.

Примеры

<code>swap.in</code>	<code>swap.out</code>
4	3
1 4	1
1 3	4
4 5	2
1 4	

Задача D. Равные подстроки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Дана строка $S = s_1 s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, равны ли подстроки $s_{l_1} \dots s_{r_1}$ и $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «+», если соответствующие подстроки равны, и «-», в противном случае.

Примеры

стандартный ввод	стандартный вывод
abacaba 4 1 1 7 7 1 3 5 7 3 4 4 5 1 7 1 7	++-+
qa 3 1 1 1 1 2 2 2 2 1 1 2 2	++-

Задача Е. Сравнение подстрок

Имя входного файла: `substrcmp.in`
Имя выходного файла: `substrcmp.out`
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

Дана строка $S = s_1 s_2 \dots s_n$ и множество запросов вида (l_1, r_1, l_2, r_2) . Для каждого запроса требуется ответить, какая из подстрок больше — $s_{l_1} \dots s_{r_1}$ или $s_{l_2} \dots s_{r_2}$.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l_1, r_1, l_2, r_2 ($1 \leq l_1 \leq r_1 \leq |S|$; $1 \leq l_2 \leq r_2 \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «=», если соответствующие подстроки равны, «>», если первая подстрока больше и «<», если первая подстрока меньше.

Примеры

substrcmp.in	substrcmp.out
abacaba	=
3	<
1 3 5 7	>
1 3 3 5	
4 7 2 5	
ab	<
2	<
1 1 2 2	
1 1 1 2	

Задача F. Подпалиндромы

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Строка называется *палиндромом*, если она одинаково читается слева направо и справа налево. Иными словами, строка $s_1s_2\dots s_k$ — палиндром, если $s_1 = s_k$, $s_2 = s_{k-1}$ и т.д.

Дана строка $S = s_1s_2\dots s_n$ и множество запросов вида (l, r) . Для каждого запроса требуется ответить, является ли подстрока $s_l\dots s_r$ палиндромом.

Формат входных данных

В первой строке записана строка S , состоящая из строчных латинских букв. Эта строка непустая и имеет длину не более 100 000 символов. Во второй строке записано целое число q ($1 \leq q \leq 100\,000$) — количество запросов. В каждой из следующих q строк записаны числа l и r ($1 \leq l \leq r \leq |S|$).

Формат выходных данных

Для каждого запроса выведите «+», если подстрока $s_l\dots s_r$ является палиндромом, и «-», в противном случае.

Примеры

стандартный ввод	стандартный вывод
abacaba	++-+
4	
1 7	
2 2	
1 5	
3 5	

Задача G. Одинаковые квадраты

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

На разборе задач одного из констестов петрозаводских сборов Вова и Саша поспорили, кто из них сможет найти за 300 минут в матрице размера $N \times M$, состоящей из строчных латинских букв, пару одинаковых квадратов наибольшего размера. Квадраты могут накладываться друг на друга, но не могут совпадать. Кто нашёл пару большего размера, тот и выиграл. Мимо проходил Петя, посмотрел на матрицу, сказал, что оптимальная пара квадратов имеет сторону K , и пошёл дальше. Вова и Саша до сих пор пытаются найти этот ответ. Может быть, вы скажете, какую пару квадратов имел в виду Петя?

Формат входных данных

В первой строке через пробел даны два целых числа N и M . $1 \leq N, M \leq 500$. В следующих N строках по M символов в каждой строке приведена матрица из строчных латинских букв.

Формат выходных данных

В первой строке выведите целое число K , которое сказал Петя. В следующих двух строках выведите координаты левой верхней клетки каждого из квадратов. Если существует более одной пары одинаковых квадратов наибольшего размера, то можно вывести любую из них. Вы можете считать, что левая верхняя клетка матрицы имеет координаты $(1, 1)$, правая нижняя — координаты (N, M) . Если Петя сказал, что в матрице не существует пары одинаковых квадратов, выведите единственное число 0.

Примеры

стандартный ввод	стандартный вывод
5 10 ljkfghdfas isdfjksiye pgljkijlgp eyisdafdsi lnpglkfkjl	3 1 1 3 3

Задача Н. Взлом хеширования

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

Ваши решения не работают на крайних случаях?
Встроенная быстрая сортировка неожиданно стала работать за квадратичное время?
В геометрических задачах не хватает точности вычислений?
Решение проходит локальное стресс-тестирование, но не работает на тестах жюри?
Именно в вашем случае ошибка оказалась не в решении, а в библиотечной функции?
Хотите узнать, кто за всем этим стоит?

Сегодня у вас есть уникальная возможность вступить в тайную организацию:
Орден Коварных Бобров! Члены этой организации делают в среднем на 146% больше
успешных взломов, чем непосвящённые, а в задачи их авторства тесты приходится
добавлять в несколько раз реже. Чтобы подать заявку на вступление, необходимо пройти
вступительное испытание: решить предложенную ниже задачу.

Торопитесь! Количество мест ограничено!

В этой задаче требуется найти коллизию при полиномиальном хешировании строк, состоящих из маленьких букв английского алфавита.

Полиномиальный хеш строки имеет два параметра: множитель p и модуль q . Для пустой строки ε значение хеш-функции $h(\varepsilon) = 0$, а для любой строки S и любого символа c хеш-функция рекуррентно определяется как $h(S + c) = (h(S) \cdot p + \text{code}(c)) \bmod q$. Здесь $\text{code}(c)$ — это ASCII-код символа c . Как известно, коды маленьких букв английского алфавита идут подряд: $\text{code}(\text{'a'}) = 97$, $\text{code}(\text{'b'}) = 98$, \dots , $\text{code}(\text{'z'}) = 122$. Можно выписать и нерекуррентную формулу: если строка $S = s_1 s_2 \dots s_n$, то $h(S) = (\text{code}(s_1) \cdot p^{n-1} + \text{code}(s_2) \cdot p^{n-2} + \dots + \text{code}(s_n) \cdot p^0) \bmod q$.

По заданным числам p и q найдите две различные непустые строки A и B такие, что $h(A) = h(B)$.

Формат входных данных

Первая строка ввода содержит два целых числа p и q , разделённых пробелом — параметры функции хеширования ($0 < p < q < 2 \cdot 10^{18}$).

Формат выходных данных

В первых двух строках выведите две различные непустые строки A и B , для которых $h(A) = h(B)$. Строки должны состоять исключительно из маленьких букв английского алфавита (ASCII-коды 97–122) и иметь длину от 1 до 100 000 символов. Заметим, что длины строк не обязательно должны совпадать. Если возможных ответов несколько, разрешается вывести любой из них.

Примеры

<code>breaking-hashing.in</code>	<code>breaking-hashing.out</code>
31 47	aa bq

Замечание

В примере $h(A) = (97 \cdot 31 + 97) \bmod 47 = 3104 \bmod 47 = 2$ и
 $h(B) = (98 \cdot 31 + 113) \bmod 47 = 3151 \bmod 47 = 2$.

Задача I. Подстроки

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

Дано K строк из маленьких латинских букв. Требуется найти их наибольшую общую подстроку.

Формат входных данных

В первой строке число K ($1 \leq K \leq 10$). В следующих K строках — собственно K строк (длины строк от 1 до 10 000).

Формат выходных данных

Наибольшая общая подстрока. Гарантируется, что она не пуста.

Примеры

<code>substr.in</code>	<code>substr.out</code>
3 abacaba mycabarchive acabistrue	cab