

## Задача А. Сравнения подстрок

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

Дана строка. Нужно уметь отвечать на запросы вида: равны ли подстроки  $[a \dots b]$  и  $[c \dots d]$ .

### Формат входных данных

В первой строке записана непустая строка  $S$ , состоящая из не более чем  $10^5$  строчных латинских букв. Во второй строке записано целое число  $M$  — количество запросов.

В следующих  $M$  строках записаны запросы. Каждый запрос задаётся четырьмя целыми числами  $a, b, c, d$ .

$$0 \leq M \leq 10^5, 1 \leq a \leq b \leq |S|, 1 \leq c \leq d \leq |S|.$$

### Формат выходных данных

Выведите  $M$  строк, по одной для каждого запроса. Выведите в соответствующей строке «Yes», если подстроки совпадают, и «No» иначе.

### Примеры

substrcmp.in	substrcmp.out
trololo	Yes
3	Yes
1 7 1 7	No
3 5 5 7	
1 1 1 5	

## Задача В. Топологическая сортировка

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

Дан ориентированный невзвешенный граф. Необходимо его топологически отсортировать.

### Формат входных данных

В первой строке входного файла даны два целых числа  $N$  и  $M$  ( $1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$ ) — количества вершин и рёбер в графе соответственно. Далее в  $M$  строках перечислены рёбра графа. Каждое ребро задаётся парой чисел — номерами начальной и конечной вершин соответственно.

### Формат выходных данных

Вывести любую топологическую сортировку графа в виде последовательности номеров вершин. Если граф невозможно топологически отсортировать, вывести «-1».

### Примеры

<code>topsort.in</code>	<code>topsort.out</code>
6 6 1 2 3 2 4 2 2 5 6 5 4 6	4 6 3 1 2 5

## Задача С. Ближайшая большая справа

Имя входного файла: `nearandmore.in`  
Имя выходного файла: `nearandmore.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Дан массив  $a$  из  $n$  чисел. Нужно обрабатывать запросы:

0. `set(i, x)` – выполнить присвоение  $a[i] = x$ ;
1. `get(i, x)` – найти  $\min k: k \geq i$  и  $a_k \geq x$ .

### Формат входных данных

На первой строке длина массива  $n$  и количество запросов  $m$ . На второй строке  $n$  целых чисел – массив  $a$ . Следующие  $m$  строк содержат запросы.

Индексы в массиве нумеруются с 1.

Запрос типа `set`: “0 i x”.

Запрос типа `get`: “1 i x”.

$$1 \leq n, m \leq 200\,000.$$

$$1 \leq i \leq n.$$

$$0 \leq x, a_i \leq 200\,000.$$

### Формат выходных данных

На каждой запрос типа `get` на отдельной строке выведите  $k$ .

Если такого  $k$  не существует, выведите  $-1$ .

### Примеры

<code>nearandmore.in</code>	<code>nearandmore.out</code>
4 5	1
1 2 3 4	3
1 1 1	-1
1 1 3	2
1 1 5	
0 2 3	
1 1 3	

## Задача D. Раскраска в три цвета

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

Петя нарисовал на бумаге  $n$  кружков и соединил некоторые пары кружков линиями. После этого он раскрасил каждый кружок в один из трех цветов — красный, синий или зеленый.

Теперь Петя хочет изменить их раскраску. А именно — он хочет перекрасить каждый кружок в некоторый другой цвет так, чтобы никакие два кружка одного цвета не были соединены линией. При этом он хочет обязательно перекрасить каждый кружок, а перекрашивать кружок в тот же цвет, в который он был раскрашен исходно, не разрешается.

Помогите Пете решить, в какие цвета следует перекрасить кружки, чтобы выполнялось указанное условие.

### Формат входных данных

Первая строка содержит два целых числа  $n$  и  $m$  — количество кружков и количество линий, которые нарисовал Петя, соответственно ( $1 \leq n \leq 1\,000$ ,  $0 \leq m \leq 20\,000$ ).

Следующая строка содержит  $n$  символов из множества  $\{\text{'R'}, \text{'G'}, \text{'B'}\}$  —  $i$ -й из этих символов означает цвет, в который раскрашен  $i$ -й кружок ( $\text{'R'}$  — красный,  $\text{'G'}$  — зеленый,  $\text{'B'}$  — синий).

Следующие  $m$  строк содержат по два целых числа — пары кружков, соединенных отрезками.

### Формат выходных данных

Выведите в выходной файл одну строку, состоящую из  $n$  символов из множества  $\{\text{'R'}, \text{'G'}, \text{'B'}\}$  — цвета кружков после перекраски. Если решений несколько, выведите любое.

Если решения не существует, выведите в выходной файл слово `Impossible`.

### Примеры

color.in	color.out
4 5 RRRG 1 3 1 4 3 4 2 4 2 3	GGBR
4 5 RGRR 1 3 1 4 3 4 2 4 2 3	Impossible

## Задача Е. 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  $\varepsilon$  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:  $\text{code}('a') = 97$ ,  $\text{code}('b') = 98$ , ...,  $\text{code}('z') = 122$ . Можно выписать и нереккуррентную 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  
 $h(B) = (98 \cdot 31 + 113) \bmod 47 = 3151 \bmod 47 = 2$ .