

## Содержание

|  |          |
|--|----------|
| <b>Задачи</b>                                | <b>2</b> |
| Задача 6А. Суффиксное дерево [1 sec, 256 mb] | 2        |
| Задача 6В. Ненокку [1 sec, 256 mb]           | 3        |
| Задача 6С. Рефрен [1 sec, 256 mb]            | 4        |
| <b>Бонусные задачи</b>                       | <b>5</b> |
| Задача 6D. Подстроки-4 [1 sec, 256 mb]       | 5        |

---

В некоторых задачах большой ввод и вывод. Пользуйтесь **быстрым вводом-выводом**.

В некоторых задачах нужен STL, который активно использует динамическую память (set-ы, map-ы) **переопределение стандартного аллокатора** ускорит вашу программу.

## Задачи

### Задача 6А. Суффиксное дерево [1 сек, 256 mb]

Дана строка  $s$ . Постройте сжатое суффиксное дерево для строки  $s$  и выведите его. Найдите такое дерево, которое содержит минимальное количество вершин.

#### Формат входных данных

В первой строке записана строка  $s$  ( $1 \leq |s| \leq 10^5$ ), последний символ строки доллар «\$», остальные символы строки маленькие латинские буквы.

#### Формат выходных данных

Пронумеруйте вершины дерева от 0 до  $n - 1$  в порядке обхода в глубину, обходя поддеревья в порядке лексикографической сортировки исходящих из вершины рёбер. Используйте ASCII-коды символов для определения их порядка.

В первой строке выведите число  $n$  – количество вершин дерева. В следующих  $n - 1$  строках выведите описание вершин дерева, кроме корня, в порядке увеличения их номеров.

Описание вершины дерева  $v$  состоит из трёх целых чисел:  $p, lf, rf$ , где  $p$  ( $0 \leq p \leq n, p \neq v$ ) – номер родителя текущей вершины. На ребер ведущем из  $p$  в  $v$  написана подстрока  $s[lf..rf)$  ( $0 \leq lf < rf \leq |s|$ ).

#### Примеры

| stdin   | stdout  |
|---------|---|
| aaa\$   | 7<br>0 3 4<br>0 0 1<br>2 3 4<br>2 1 2<br>4 3 4<br>4 2 4                             |
| b\$     | 3<br>0 1 2<br>0 0 2   |
| ababa\$ | 10<br>0 5 6<br>0 0 1<br>2 5 6<br>2 1 3<br>4 5 6<br>4 3 6<br>0 1 3<br>7 5 6<br>7 3 6 |

### Задача 6В. Ненокку [1 sec, 256 mb]

Очень известный автор не менее известной книги решил написать продолжение своего произведения. Он писал все свои книги на компьютере, подключенном к интернету. Из-за такой неосторожности мальчику Ненокку удалось получить доступ к еще ненаписанной книге. Каждый вечер мальчик залазил на компьютер писателя и записывал на свой компьютер новые записи. Ненокку, записав на свой компьютер очередную главу, заинтересовался, а использовал ли хоть раз писатель слово “книга”. Но он не любит читать книги (он лучше ползает в интернете), и поэтому он просит вас узнать есть ли то или иное слово в тексте произведения. Но естественно его интересует не только одно слово, а достаточно много.

#### Формат входных данных

В каждой строчке входного файла записано одна из двух записей.

1. ? <слово> (<слово> — это набор не более 50 латинских символов);
2. A <текст> (<текст> — это набор не более  $10^5$  латинских символов).

1 означает просьбу проверить существование подстроки <слово> в произведение.

2 означает добавление в произведение <текст>.

Писатель только начал работать над произведением, поэтому он не мог написать более  $10^5$  символов. Суммарная длина всех запросов не превосходит 15 мегабайт плюс 12140 байт.

#### Формат выходных данных

Выведите на каждую строчку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

#### Пример

| stdin       | stdout |
|-------------|--------|
| ? love      | NO     |
| ? is        | NO     |
| A Loveis    | YES    |
| ? love      | NO     |
| ? WHO       | YES    |
| A Whoareyou |        |
| ? is        |        |

### Задача 6С. Рефрен [1 sec, 256 mb]

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 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.

#### Пример

| stdin             | stdout |
|-------------------|--------|
| 9 3               | 9      |
| 1 2 1 2 1 3 1 2 1 | 3      |
|                   | 1 2 1  |

## Бонусные задачи

### Задача 6D. Подстроки-4 [1 sec, 256 mb]

Даны  $K$  строк из маленьких латинских букв. Найдите их наибольшую общую подстроку.

#### Формат входных данных

В первой строке число  $K$  ( $1 \leq K \leq 10$ ). Далее  $K$  строк длины от 1 до 200 000.

#### Формат выходных данных

Наибольшая общая подстрока.

#### Примеры

| stdin                                      | stdout |
|--|--------|
| 3<br>abacaba<br>mycabarchive<br>acabistrue | cab    |