

## Содержание

<b>Обязательные задачи</b>	<b>3</b>
Задача 26А. Словарь [0.6 sec, 266 mb]	3
Задача 26В. Поиск набора образцов [0.6 sec, 266 mb]	4
Задача 26С. К-я строка [0.3 sec, 266 mb]	5
Задача 26D. Prof. Жужжащий профессор [0.5 sec, 266 mb]	6
Задача 26Е. Свобода выбора [0.6 sec, 266 mb]	7
<b>Бонус</b>	<b>8</b>
Задача 26F. Подстроки-3 [0.2 sec, 266 mb]	8
Задача 26G. Вирусы [0.2 sec, 266 mb]	9
Задача 26H. Набор строк [3.0 sec, 266 mb]	10

---

### Пример работы с файлами.

Если вы не умеете читать/выводить данные, или открывать файлы, воспользуйтесь примерами. <http://acm.math.spbu.ru/~sk1/algo/sum/>

### Пример работы с файлами.

В некоторых задачах большой ввод и вывод. Про ввод-вывод в C++:

[http://acm.math.spbu.ru/~sk1/algo/input-output/cpp\\_common.html](http://acm.math.spbu.ru/~sk1/algo/input-output/cpp_common.html)

Имеет смысл пользоваться супер быстрым вводом-выводом. Две версии:

[http://acm.math.spbu.ru/~sk1/algo/input-output/io\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/io_export.cpp.html)

[http://acm.math.spbu.ru/~sk1/algo/input-output/fread\\_write\\_export.cpp.html](http://acm.math.spbu.ru/~sk1/algo/input-output/fread_write_export.cpp.html)

### Выделение памяти.

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

<http://acm.math.spbu.ru/~sk1/algo/memory.cpp.html>

## Обязательные задачи

### Задача 26А. Словарь [0.6 sec, 266 mb]

Дан набор слов и текст, требуется определить для каждого слова, присутствует ли оно в тексте как подстрока.

#### Формат входных данных

В первой строке дан текст (не более  $10^6$  строчных латинских букв). Далее дано число  $M$  — количество слов в словаре.

В следующих  $M$  строках записаны слова (не более 30 строчных латинских букв). Слова различны и отсортированы в лексикографическом порядке.

Суммарная длина слов в словаре не более  $10^5$ .

#### Формат выходных данных

$M$  строк вида Yes, если слово присутствует, и No иначе.

#### Пример

dictionary.in	dictionary.out
trololo	No
3	Yes
abacabadabacaba	Yes
olo	
trol	

#### Замечание

Эта задача решается просто Бором. Без алгоритма Ахо-Корасик.

### Задача 26В. Поиск набора образцов [0.6 sec, 266 mb]

Напишите программу, которая для каждой строки из заданного набора  $S$  проверяет, верно ли, что она содержит как подстроку одну из строк из набора  $T$ .

#### Формат входных данных

Первая строка входного файла содержит натуральное число  $n$  — количество строк в наборе  $T$  ( $n \leq 1000$ ). Каждая из следующих  $n$  строк содержит непустую строку длины не более 80-ти символов.

Оставшаяся часть файла содержит строки из набора  $S$ . Каждая строка состоит из ASCII символов с кодами от 32 до 126 включительно. Строка может быть пустой и ее длина не превышает 250-ти символов.

Гарантируется, что размер входного файла не превышает 1 Мбайт.

#### Формат выходных данных

В выходной файл выведите все строки из набора  $S$  (в том порядке, в котором они находятся во входном файле), содержащие как подстроку по крайней мере одну строку из набора  $T$ .

#### Пример

console.in	console.out
3 gr sud abc lksh sudislavl kostroma summer group a'	sudislavl group a'

#### Замечание

Если у вас WA 36, вы неправильно читаете входные данные. Строки могут состоять только из пробелов.

### Задача 26С. К-я строка [0.3 сек, 266 mb]

Реализуйте структуру данных, которая поддерживает следующие операции:

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

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

kthstr.in	kthstr.out
7	tolstoy
pushkin	gogol
lermontov	
tolstoy	
gogol	
gorkiy	
5	
1	

### Задача 26D. Prof. Жужжащий профессор [0.5 сек, 266 mb]

В одном очень известном университете один очень известный профессор очень быстро произносил свои лекции, так, что ничего невозможно было разобрать. Студенты шутили по этому поводу, что он не говорит, а жужжит. Естественно, что про загадочного профессора никто абсолютно ничего не знал.

Но вот недавно Петя Булочкин решил предпринять исследование по изучению словарного запаса профессора. С этой целью он даже посетил одну лекцию и записал все сказанное на ней на диктофон. Затем, прокручивая дома запись с десятикратным замедлением, Петя смог записать все, что сказал профессор. Но вот незадача — профессор говорил так быстро, что, даже прослушивая замедленную запись, нельзя было точно сказать, где он делал паузы между словами. Таким образом, у Пети есть некоторый текст  $S$ , состоящий только из маленьких латинских букв — лекция, которая была прочитана профессором.

Петя решил, что те слова, которые профессор употреблял только один раз во время своей лекции, его не интересуют. Кроме того, понятно, что если профессор употреблял некоторое слово два или более раз, то существуют два неперекрывающихся вхождения этого слова в текст  $S$ . Назовем непустую строку  $T$  кандидатом в слова, если существуют два неперекрывающихся вхождения  $T$  в  $S$ . Теперь Петя хочет найти все строки, которые являются кандидатами в слова. И поможете ему в этом Вы.

#### Формат входных данных

Единственная строка входного файла содержит от 1 до 3000 маленьких латинских букв. Это и есть текст  $S$ , который прочитал профессор на лекции.

#### Формат выходных данных

Единственная строка выходного файла должна содержать одно число, равное количеству строк, являющихся кандидатами в слова.

#### Пример

prof.in	prof.out
bbaabbbabb	7

#### Замечание

Решение за  $\mathcal{O}(n^2)$  получит ОК. А вот с  $\mathcal{O}(n^2)$  памяти — нет.

**Задача 26Е. Свобода выбора [0.6 сек, 266 mb]**

Даны две строки, состоящих из заглавных латинских букв. Нужно найти их наибольшую общую подстроку. Полное условие можно посмотреть на [тимусе](#).

**Формат входных данных**

На первой строке число  $n$  ( $1 \leq n \leq 10^5$ ).

На второй и третьей строках находятся по  $n$  заглавных английских букв.

**Формат выходных данных**

Максимальную по длине общую подстроку. Если оптимальных ответов несколько, выведите любой.

**Примеры**

freedom.in	freedom.out
28 VOTEFORTHEGREATALBANIAFORYOU CHOOSETHEGREATALBANIANFUTURE	THEGREATALBANIA

**Замечание**

19-й – антихеш тест

## Бонус

### Задача 26F. Подстроки-3 [0.2 sec, 266 mb]

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

#### Формат входных данных

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

#### Формат выходных данных

Наибольшая общая подстрока.

#### Примеры

substr3.in	substr3.out
3 abacaba mycabarchive acabistrue	cab

#### Замечание

Почти такая же, как “freedom”.



### Задача 26G. Вирусы [0.2 sec, 266 mb]

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 сегмент (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.

#### Пример

Для множества кодов {011, 11, 0000} примером бесконечной безопасной последовательности является 010101... Для множества {01, 11, 00000} бесконечной безопасной последовательности не существует.

#### Формат входных данных

Первая строка входного файла `virus.in` 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 30 000.

#### 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.

#### Пример

<code>virus.in</code>	<code>virus.out</code>
3 01 11 00000	NIE
3 011 11 0000	ТАК

### Задача 26Н. Набор строк [3.0 sec, 266 mb]

В Инновационном Отделе НИИ Исследований Данных Строк разработана клавиатура для внутреннего пользования, облегчающая набор строк огромной длины. Кроме обычных клавиш, соответствующих маленьким латинским буквам, на клавиатуре есть еще  $n$  функциональных клавиш  $F_1, \dots, F_n$ , соответствующих заданным строкам из словаря  $S_1, \dots, S_n$ . При нажатии такой клавиши  $F_i$  строка  $S_i$  загружается во внутреннюю память клавиатуры. В каждый момент времени в памяти может находиться не более одной строки из словаря.

Кроме того, в клавиатуру встроены графический манипулятор «Кыш», с помощью которого легким движением руки можно ввести любую подстроку находящейся в памяти строки.

Вася занимается исследованием эффективности данного нововведения. Для этого ему требуется написать программу, которая будет вычислять минимальное необходимое количество действий (нажатий и использований «Кыш») для ввода данной строки  $S$ . В момент начала ввода строки память пуста.

Например, если требуется ввести строку “abacaba”, а в словаре есть строки “baba” и “caca”, то это можно сделать за четыре действия — нажать  $F_1$ , выбрать манипулятором подстроку “aba”, затем нажать ‘с’, и опять выбрать манипулятором подстроку “aba”. Если бы нужно было набрать с таким словарем “bacababa”, то это можно сделать за пять действий: ‘b’,  $F_2$ , “aca”,  $F_1$ , “baba”.

#### Формат входных данных

В первой строке входного файла задано число  $n$  ( $1 \leq n \leq 50$ ). В последующих  $n$  строках заданы  $S_i$ , составленные из не более чем 500 символов. В последней строке вводится непустая строка  $S$ , длина которой не превосходит 100 000. Все символы строк — маленькие латинские буквы.

#### Формат выходных данных

Выведите минимальное необходимое количество действий.

#### Пример

typing.in	typing.out
2 baba caca abacaba	4
2 baba caca bacababa	5

#### Замечание

Можно получить ОК без быстрых суффструктур.