

## Задача А. Циклические сдвиги

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

$k$ -м *циклическим сдвигом* строки  $S$  называется строка, полученная перестановкой  $k$  первых символов строки  $S$  в конец строки.

Рассмотрим все различные циклические сдвиги строки  $S$  и отсортируем их по возрастанию. Требуется вычислить  $i$ -ю строчку этого массива.

Например, для строки `abacabac` существует четыре различных циклических сдвига: нулевой (`abacabac`), первый (`bacabaca`), второй (`acabacab`) и третий (`cabacaba`). После сортировки по возрастанию получится такой массив: `abacabac`, `acabacab`, `bacabaca`, `cabacaba`.

### Формат входных данных

В первой строке входного файла записана строка  $S$ , длиной не более 100 000 символов с ASCII-кодами от 32 до 126. Во второй строке содержится единственное целое число  $k$  ( $1 \leq k \leq 100\,000$ ).

### Формат выходных данных

В выходной файл выведите  $k$ -й по возрастанию циклический сдвиг строки  $S$ , или слово IMPOSSIBLE, если такого сдвига не существует.

### Пример

стандартный ввод	стандартный вывод
abacabac 4	cabacaba

## Задача В. Суффиксный массив (2 балла)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Постройте суффиксный массив для заданной строки  $s$ .

### Формат входных данных

Первая строка входного файла содержит строку  $s$  ( $1 \leq |s| \leq 400\,000$ ). Строка состоит из строчных латинских букв.

### Формат выходных данных

Выведите  $|s|$  различных чисел — номера первых символов суффиксов строки  $s$  так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке.

### Пример

стандартный ввод	стандартный вывод
ababb	1 3 5 2 4

## Задача С. Просто LCP

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Дана строка длины  $N$  и отсортированный массив суффиксов этой строки (т.е. суффиксный массив), вам нужно вычислить LCP. При сортировке строка **a** считается меньше строки **aa**. LCP — наибольший общий префикс двух последовательных суффиксов в суффиксном массиве.

### Формат входных данных

В первой строке число  $N$  ( $1 \leq N \leq 10^5$ ). На второй строке файла дана  $N$  строчных латинских букв. В третьей строке  $N$  чисел от 1 до  $N$  — суффиксный массив (числом  $i$  кодируется суффикс, начинающийся с  $i$ -го символа).

### Формат выходных данных

Выведите  $N - 1$  число — значения LCP.

### Примеры

стандартный ввод	стандартный вывод
5 сасао 2 4 1 3 5	1 0 2 0
3 bab 2 3 1	0 1

### Замечание

Суффиксный массив для строки сасао:

асао  
ао  
сасао  
сао  
о

## Задача D. Задача про негра

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	16 мегабайт

Как известно, у негров бывают очень длинные и сложные имена. Кроме того, ходят слухи, что эти имена генерируются случайно (<https://www.youtube.com/watch?v=vJJEVBMtOEQ>).



Напишите алгоритм, который вычисляет суффиксный массив заданного имени, используя  $O(1)$  дополнительной памяти.

Чтобы не получить ML, ваш код может выглядеть так:

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    int n; scanf("%d", &n);
    char* s = (char*)malloc(n + 1);
    scanf("%s", s);
    int *res = (int*)malloc((n + 1) * sizeof(*res));
    /*
    your solution here...
    */
    for (int i = 0; i < n; ++i)
        printf("%d ", res[i]);
    printf("%c", 10);
}
```

### Формат входных данных

В первой строке дано целое число  $n$  ( $1 \leq n \leq 2 \cdot 10^6$ ) — длина имени негра.

Во второй строке записано имя негра, состоящее из строчных букв латинского алфавита.

Гарантируется, что имя негра сгенерировано случайно.

### Формат выходных данных

Выведите  $n$  целых чисел — суффиксный массив имени негра.

## Примеры

стандартный ввод	стандартный вывод
12 uvuvwevwevwe	12 9 6 1 3 2 10 7 4 11 8 5
12 onyetenyeve	12 6 4 9 2 7 1 5 10 11 3 8
10 ugwemuhwem	9 4 2 7 10 5 1 6 8 3
4 osas	3 1 4 2

## Задача Е. Различные подстроки

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

Дана строка  $S$ . Назовем ее подстрокой строку с  $i$ -го по  $j$ -й символ ( $i \leq j$ ). Ваша задача — посчитать количество различных подстрок данной строки.

### Формат входных данных

Во входном файле находится одна строка  $S$ , состоящая не более, чем из 200 000 символов. Все символы в строке — маленькие латинские буквы.

### Формат выходных данных

В выходной файл выведите единственное число — количество различных подстрок заданной строки.

### Пример

стандартный ввод	стандартный вывод
aaba	8

## Задача F. Контрольное списывание (3 балла)

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

Сегодня на уроке преподаватель Массивов Автомат Укконеви́ч рассказывал своим ученикам про строки, суффиксные структуры и всё такое. Например, он рассказал им, как сравнить две строки  $A$  и  $B$  *лексикографически*. Если одна из них является префиксом другой, то более короткая будет лексикографически меньше, иначе необходимо сравнить символы стоящие на первой позиции, в которой они отличаются. Строка с меньшим по номеру в алфавите символом на данной позиции и будет лексикографически меньше.

Чтобы проверить понимание учениками нового материала, Автомат Укконеви́ч дал им следующее задание: найти  $k$ -ю лексикографически непустую уникальную подстроку строки  $S$ .

Так как учитель знает, что Александр Г. и Илья С. очень любят списывать у известного в узких кругах Демида Г., каждый школьник получил своё число  $k$  и вынужден был обратиться к вам за помощью.

### Формат входных данных

В первой строке входного файла находится строка  $S$  ( $|S| \leq 10^5$ ). Вторая строка содержит число  $k$  ( $1 \leq k \leq 10^{18}$ ) — порядковый номер запрашиваемой подстроки.

### Формат выходных данных

Если ответ существует, выведите искомую подстроку строки  $S$ . В противном случае выведите её лексикографически максимальную подстроку.

### Пример

стандартный ввод	стандартный вывод
abacaba 10	acab

## Задача G. Рефрен

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

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 входного файла 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

стандартный ввод	стандартный вывод
9 3	9
1 2 1 2 1 3 1 2 1	9
	1 2 1 2 1 3 1 2 1



## Задача Н. Палиндромы

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 0.5 секунд  
Ограничение по памяти: 256 мегабайт

Вам дана строка  $s$  из маленьких английских букв. Определим для подстроки *красоту* как количество вхождений подстроки в строку, умноженное на длину подстроки. Для данной строки найдите максимальную красоту среди всех её палиндромных подстрок.

### Формат входных данных

Первая строка содержит строку  $s$ . Длина строки не превосходит  $3 \cdot 10^5$ .

### Формат выходных данных

Выведите одно число — ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
abacaba	7
www	4

## Задача I. Ключ к шифру (3 балла)

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Сейчас Эркюль Пуаро занят разоблачением международного преступного синдиката, занимающегося контрабандой предметов искусства. Полиция, сотрудничающая с Пуаро, перехватила зашифрованное письмо, содержащее информацию о месте и времени предстоящей сделки, на которой будет присутствовать и глава синдиката. Чтобы сорвать сделку и задержать главу синдиката, необходимо расшифровать перехваченное письмо.

Эркюль знает, что ключ к шифру вычисляется из строки  $s$ . Обозначим за  $f(w)$  длину максимального суффикса  $w$ , не равного  $w$ , который является и префиксом  $w$ . Например,  $f(abc) = 0$ ,  $f(abab) = 2$ ,  $f(aaa) = 2$ . Тогда ключом является максимум по всем  $t$ , являющимся подстроками  $s$ , величины  $(|t| + f(t)^2)$ . Помогите Эркюлю вычислить ключ.

### Формат входных данных

В единственной строке дана строка  $s$ , состоящая из строчных латинских букв ( $1 \leq |s| \leq 500\,000$ ).

### Формат выходных данных

Выведите единственное целое число — искомый ключ к шифру.

### Пример

стандартный ввод	стандартный вывод
ababaab	14

## Задача J. Тандемные повторы

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

Дана строка  $s$  длины  $n$ .

Тандемным повтором в ней называются два вхождения какой-либо подстроки подряд. Иными словами, тандемный повтор описывается парой индексов  $i < j$  такими, что подстрока  $s[i \dots j]$  — это две одинаковые строки, записанные подряд.

От вас требуется посчитать количество пар индексом  $i < j$  таких, что подстрока  $s[i \dots j]$  является тандемным повтором.

### Формат входных данных

Во входном файле находятся не более 30 тестов. Каждый тест состоит из единственной непустой строки, состоящей из символов **A,C,G,T**. Длина строки не превосходит  $10^5$ . Входной файл заканчивается строкой **0**.

### Формат выходных данных

Для каждого теста выведите единственное число — количество тандемных повторов. Числа разделяйте переводами строк.

### Пример

стандартный ввод	стандартный вывод
AGGA	1
AGAG	1
ATTTCGATTTCGATTTCG	9
AAAA	4
0	

## Задача К. Ненокку

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 64 мегабайта

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 строке 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 строку типа 1 “YES”, если существует подстрока <слово>, и “NO” в противном случае. Не следует различать регистр букв.

### Пример

стандартный ввод	стандартный вывод
? love	NO
? is	NO
A Loveis	YES
? love	NO
? WHO	YES
A Whoareyou	
? is	

## Задача L. Поиск подстроки в подстроке

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 3.5 секунд  
Ограничение по памяти: 512 мегабайт

Опытным участникам соревнований по спортивному программированию хорошо известна классическая задача о нахождении количества вхождений подстроки в строку. Обычно она формулируется так: дана строка-образец  $s$  и строка  $t$ , требуется найти количество индексов, начиная с которых строка  $s$  содержится в строке  $t$ .

К сожалению, для решения этой задачи уже придумано множество алгоритмов, поэтому сама по себе она может быть интересна только в качестве упражнения, но не олимпиадной задачи. Однако, как это часто бывает со стандартными задачами, её легко усложнить — представим, что нас интересуют не сами строки  $s$  и  $t$ , а некоторые их подстроки  $s[l_1 \dots r_1]$  и  $t[l_2 \dots r_2]$ .

Как вы уже, наверное, догадались, вам дано  $q$  запросов,  $i$ -й из которых задаёт некоторые подстроки  $\bar{s} = s[l_{1i} \dots r_{1i}]$  и  $\bar{t} = t[l_{2i} \dots r_{2i}]$ . Для каждого такого запроса необходимо посчитать количество вхождений строки  $\bar{s}$  в строку  $\bar{t}$ .

### Формат входных данных

Первая и вторая строки входных данных содержат строки  $s$  и  $t$  ( $1 \leq |s|, |t| \leq 2 \cdot 10^5$ ) соответственно. Строки состоят из маленьких букв английского алфавита.

В третьей строке задано одно число  $q$  ( $1 \leq q \leq 5 \cdot 10^5$ ) — количество запросов.

Каждая из следующих  $q$  строк содержит по четыре числа  $l_1, r_1, l_2$  и  $r_2$  ( $1 \leq l_1 \leq r_1 \leq |s|, 1 \leq l_2 \leq r_2 \leq |t|$ ), описывающих очередной запрос.

### Формат выходных данных

Выведите  $q$  чисел — ответы на запросы.

### Пример

стандартный ввод	стандартный вывод
abb	3
ababababb	1
5	2
1 2 1 7	1
2 3 2 9	4
3 3 4 7	
1 2 2 4	
1 1 1 9	

### Замечание

Рассмотрим запросы в первом примере. Для индексации позиций вхождения будем использовать изначальные позиции в строке  $t$ .

- $\bar{s} = ab, \bar{t} = abababa$ .  $\bar{s}$  входит в  $\bar{t}$ , начиная с индексов  $[1, 3, 5]$ .
- $\bar{s} = bb, \bar{t} = babababb$ .  $\bar{s}$  входит в  $\bar{t}$ , начиная с индекса  $[8]$ .
- $\bar{s} = b, \bar{t} = baba$ .  $\bar{s}$  входит в  $\bar{t}$ , начиная с индексов  $[4, 6]$ .
- $\bar{s} = ab, \bar{t} = bab$ .  $\bar{s}$  входит в  $\bar{t}$ , начиная с индекса  $[3]$ .
- $\bar{s} = a, \bar{t} = ababababb$ .  $\bar{s}$  входит в  $\bar{t}$ , начиная с индексов  $[1, 3, 5, 7]$ .