

А. Сравнения подстрок

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: стандартный ввод

output: стандартный вывод

Дана строка s . Ответьте на m запросов вида: равны ли подстроки $s[a..b]$ и $s[c..d]$.

Входные данные:

В первой строке ввода записана строка s ($1 \leq |s| \leq 10^5$).

Во второй строке записано целое число m — количество запросов ($0 \leq m \leq 10^5$).

В следующих m строках четверки чисел a, b, c, d ($1 \leq a \leq b \leq |s|, 1 \leq c \leq d \leq |s|$).

Выходные данные:

Выведите m строк. Выведите Yes, если подстроки совпадают, и No иначе.

Пример:

входные данные:

```
trololo
3
1 7 1 7
3 5 5 7
1 1 1 5
```

выходные данные:

```
Yes
Yes
No
```

В. Префикс-функция

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: стандартный ввод

output: стандартный вывод

Постройте префикс-функцию для заданной строки s .

Входные данные:

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

выходные данные:

Выведите значения префикс-функции строки s для всех индексов $1, 2, \dots, |s|$.

Пример:

входные данные:

aaaAAA

выходные данные:

0 1 2 0 0 0

С. Z-функция

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: стандартный ввод

output: стандартный вывод

Постройте Z-функцию для заданной строки s .

Входные данные:

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

Выходные данные:

Выведите значения Z-функции строки s для индексов $2, 3, \dots, |s|$.

Примеры:

входные данные:

```
aaaAAA
```

выходные данные:

```
2 1 0 0 0
```

входные данные:

```
abacaba
```

выходные данные:

```
0 1 0 3 0 1
```

D. Быстрый поиск подстроки в строке

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: стандартный ввод

output: стандартный вывод

Даны строки p и t . Требуется найти все вхождения строки p в строку t в качестве подстроки.

Входные данные:

Первая строка входного файла содержит p , вторая — t ($1 \leq |p|, |t| \leq 10^6$). Строки состоят из букв латинского алфавита.

Выходные данные:

В первой строке выведите количество вхождений строки p в строку t . Во второй строке выведите в возрастающем порядке номера символов строки t , с которых начинаются вхождения p . Символы нумеруются с единицы.

Пример:

входные данные:

```
aba
abaCaba
```

выходные данные:

```
2
1 5
```

Е. Поиск периода

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: стандартный ввод

output: стандартный вывод

Дана строка s . Требуется найти минимальную по длине строку t , такую что s представима в виде конкатенации одной или нескольких строк t .

Входные данные:

Первая строка входного файла содержит s ($1 \leq |s| \leq 10^6$). Строка состоит из букв латинского алфавита.

Выходные данные:

Выведите длину искомой строки t .

Примеры:

входные данные:

abcabcabc

выходные данные:

3

входные данные:

abacaba

выходные данные:

7

Г. Подстроки-3

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: стандартный ввод

output: стандартный вывод

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

Входные данные:

В первой строке число K ($1 \leq K \leq 10$).

В следующих K строках — собственно K строк (длины строк от 1 до 10000).

Выходные данные:

Наибольшая общая подстрока.

Пример:

входные данные:

```
3
abacaba
mycabarchive
acabistrue
```

выходные данные:

```
cab
```

G. Множественный поиск

time limit per test: 3 seconds

memory limit per test: 1024 megabytes

input: стандартный ввод

output: стандартный вывод

Дан массив строк s_i и строка t . Требуется для каждой строки s_i определить, встречается ли она в t как подстрока.

Входные данные:

Первая строка входного файла содержит целое число n — число элементов в s ($1 \leq 10^6$). Следующие n строк содержат по одной строке s_i . Сумма длин всех строк из s не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$). Все строки состоят из строчных латинских букв.

Выходные данные:

Для каждой строки s_i выведите «YES», если она встречается в t и «NO» в противном случае. Строки нумеруются в порядке появления во входном файле.

Пример:

входные данные:

```
3
abc
abcdr
abcde
xabcdef
```

выходные данные:

```
YES
NO
YES
```

Н. Множественный поиск 2

time limit per test: 3 seconds

memory limit per test: 1024 megabytes

input: стандартный ввод

output: стандартный вывод

Дан массив строк s_i и строка t . Требуется для каждой строки s_i определить, сколько раз она встречается в t как подстрока.

Входные данные:

Первая строка входных данных содержит целое число n ($1 \leq n \leq 10^6$) — число элементов в s .

Следующие n строк содержат по одной строке s_i . Гарантируется, что сумма длин всех строк из s не превосходит 10^6 .

Последняя строка входных данных содержит строку t ($1 \leq |t| \leq 10^6$).

Все строки состоят из строчных латинских букв.

Выходные данные:

Для каждой строки s_i выведите количество её вхождений в строку t в том же порядке, что и во входных данных.

Пример:

входные данные:

```
3
abc
abcdr
abcde
xabcdef
```

выходные данные:

```
1
0
1
```


I. Множественный поиск 3

time limit per test: 3 seconds

memory limit per test: 1024 megabytes

input: стандартный ввод

output: стандартный вывод

Дан массив строк s_i и строка t . Требуется для каждой строки s_i найти самое левое и самое правое вхождение в t как подстроки.

Входные данные:

Первая строка входного файла содержит целое число n — число элементов в s ($1 \leq n \leq 10^6$). Следующие n строк содержат по одной строке s_i . Сумма длин всех строк из s не превосходит 10^6 . Последняя строка входного файла содержит t ($1 \leq t \leq 10^6$). Все строки состоят из строчных латинских букв.

Выходные данные:

Для каждой строки s_i выведите два числа: индексы самой левой и самой правой позиции, в которых она встречается в t . Если строка не встречается в t ни разу, выведите -1 -1. Строки нумеруются в порядке появления во входном файле. Позиции нумеруются с 0.

Пример:

входные данные:

```
3
ab
bcd
abde
abcdab
```

выходные данные:

```
0 4
1 1
-1 -1
```

Ж. Суффиксный массив

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: стандартный ввод

output: стандартный вывод

Постройте суффиксный массив для заданной строки s , для каждой двух соседних суффиксов найдите длину максимального общего префикса.

Входные данные:

Первая строка входного файла содержит строку s ($1 \leq |s| \leq 400000$). Строка состоит из строчных латинских букв.

Выходные данные:

В первой строке выведите $|s|$ различных чисел — номера первых символов суффиксов строки s так, чтобы соответствующие суффиксы были упорядочены в лексикографически возрастающем порядке. Во второй строке выведите $|s| - 1$ чисел — длины наибольших общих префиксов.

Пример:

входные данные:

```
ababb
```

выходные данные:

```
1 3 5 2 4
2 0 1 1
```

К. Количество подстрок

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: стандартный ввод

output: стандартный вывод

Вычислите количество различных подстрок строки s .

Входные данные:

Единственная строка входного файла содержит строку s ($1 \leq |s| \leq 400000$). Строка состоит из строчных латинских букв.

Выходные данные:

Выведите одно число — ответ на задачу.

Примеры:

входные данные:

```
ababb
```

выходные данные:

```
11
```

L. Циклические сдвиги

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: стандартный ввод

output: стандартный вывод

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`.

Входные данные:

В первой строке входного файла записана строка S , длиной не более 100000 символов с ASCII-кодами от 32 до 126. Во второй строке содержится единственное целое число k ($1 \leq k \leq 100000$).

Выходные данные:

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

Примеры:

входные данные:

```
abacabac
4
```

выходные данные:

```
cabacaba
```

входные данные:

```
abacabac
5
```

выходные данные:

```
IMPOSSIBLE
```

М. Наибольшая общая подстрока

time limit per test: 2 seconds

memory limit per test: 512 megabytes

input: стандартный ввод

output: стандартный вывод

Найдите наибольшую общую подстроку строк s и t .

Входные данные:

Первая строка входного файла содержит строку s , вторая — t ($1 \leq |s|, |t| \leq 100,000$). Строки состоят из строчных латинских букв.

Выходные данные:

Выведите одну строку — наибольшую общую подстроку строк s и t . В случае, если ответ не единственный, выведите минимальный лексикографически.

Пример:

входные данные:

```
bababb  
zabacabba
```

выходные данные:

```
aba
```