

Яндекс.Практикум, Алгоритмы: Спринт 8. Финальные задачи

Положение участников Задачи Посылки

A. Packed Prefix

A. Packed Prefix

В. Шпаргалка

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.5 секунд	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Node.js 14.15.5	0.5 секунд	128Mb		
Python 3.7.3	2 секунды	64Mb		
Mono C# 5.2.0	0.5 секунд	128Mb		
Oracle Java 8	0.5 секунд	128Mb		
OpenJDK Java 11	0.5 секунд	128Mb		
Node JS 8.16	0.5 секунд	128Mb		

Вам даны строки в запакованном виде. Определим запакованную строку (ЗС) рекурсивно. Строка, состоящая только из строчных букв английского алфавита является ЗС. Если A и B — корректные ЗС, то и AB является ЗС. Если A — ЗС, а n — однозначное натуральное число, то $n[A]$ тоже ЗС. При этом запись $n[A]$ означает, что при распаковке строка A записывается подряд n раз. Найдите наибольший общий префикс распакованных строк и выведите его (в распакованном виде).

Иными словами, пусть сложение — это конкатенация двух строк, а умножение строки на число — повтор строки соответствующее число раз. Пусть функция f умеет принимать ЗС и распаковывать её. Если ЗС D имеет вид $D=AB$, где A и B тоже ЗС, то $f(D) = f(A) + f(B)$. Если $D=n[A]$, то $f(D) = f(A) \times n$.

Формат ввода

В первой строке записано число n ($1 \leq n \leq 1000$) — число строк. Далее в n строках записаны запакованные строки. Гарантируется, что эти строки корректны, то есть удовлетворяют указанному рекурсивному определению. Длина строк после распаковки не превосходит 10^5 .

Формат вывода

Выведите наибольший общий префикс распакованных строк.

Пример 1

Ввод

Вывод

```
3
2[a]2[ab]
3[a]2[r2[t]]
a2[aa3[b]]
```

```
aaa
```

Пример 2

Ввод

Вывод

```
3
abacabaca
2[abac]a
3[aba]
```

```
aba
```

Примечания

Сложение подразумевается как конкатенация двух строк. Умножение строки на число — повтор строки соответствующее число раз. Пусть функция f умеет принимать ЗС и распаковывать её. Если ЗС D имеет вид $D=AB$, где A и B тоже ЗС, то $f(D) = f(A) + f(B)$. Если $D = n[A]$, то $f(D) = f(A) \times n$.

А. Разворот строки

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.1 секунда	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.6	0.3 секунды	64Mb		
Mono C# 5.2.0	0.3 секунды	64Mb		
Oracle Java 8	0.3 секунды	64Mb		
OpenJDK Java 11	0.3 секунды	64Mb		

В некоторых языках предложения пишутся и читаются не слева направо, а справа налево. Вам под руку попался странный текст — в нём обычный (слева направо) порядок букв в словах. А вот сами слова идут в противоположном направлении. Вам надо преобразовать текст так, чтобы слова в нём были написаны слева направо.

Формат ввода

На ввод подаётся строка, состоящая из слов, разделённых пробелами (один пробел между соседними словами). Всего слов не более 1000, длина каждого из них — от 1 до 100 символов. Слова состоят из строчных букв английского алфавита.

Формат вывода

Выведите строку с обратным порядком слов в ней.

Пример 1

Ввод

one two three

Выход

three two one

Пример 2

Ввод

hello

Выход

hello

Пример 3

Ввод

may the force be with you

Выход

you with be force the may

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции

В. Пограничный контроль

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.2 секунды	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Mono C# 5.2.0	0.4 секунды	64Mb		
Oracle Java 8	0.4 секунды	64Mb		
OpenJDK Java 11	0.4 секунды	64Mb		

Представьте, что вы работаете пограничником и постоянно проверяете документы людей по записи из базы. При этом допустима ситуация, когда имя человека в базе отличается от имени в паспорте на одну замену, одно удаление или одну вставку символа. Если один вариант имени может быть получен из другого удалением одного символа, то человека пропустят через границу. А вот если есть какое-либо второе изменение, то человек грустно поедет домой или в посольство.

Например, если первый вариант — это «Лена», а второй — «Лера», то девушку пропустят. Также человека пропустят, если в базе записано «Коля», а в паспорте — «оля».

Однако вариант, когда в базе числится «Иннокентий», а в паспорте написано «ннакентий», уже не сработает. Не пропустят также человека, у которого в паспорте записан «Иинннокентий», а вот «Иннокентий» спокойно пересечёт границу.

Напишите программу, которая сравнивает имя в базе с именем в паспорте и решает пропускать человека или нет. В случае равенства двух строк — путешественника, естественно, пропускают.

Формат ввода

В первой строке дано имя из паспорта.
Во второй строке — имя из базы.

Обе строки состоят из строчных букв английского алфавита. Размер каждой строки не превосходит 100 000 символов.

Формат вывода

Выведите «OK», если человека пропустят, или «FAIL» в противном случае.

Пример 1

Ввод

abcdefg
abdefg

Вывод

OK

Пример 2

Ввод

helo
hello

Вывод

OK

Пример 3

Ввод

dog
fog

Вывод

OK

Пример 4

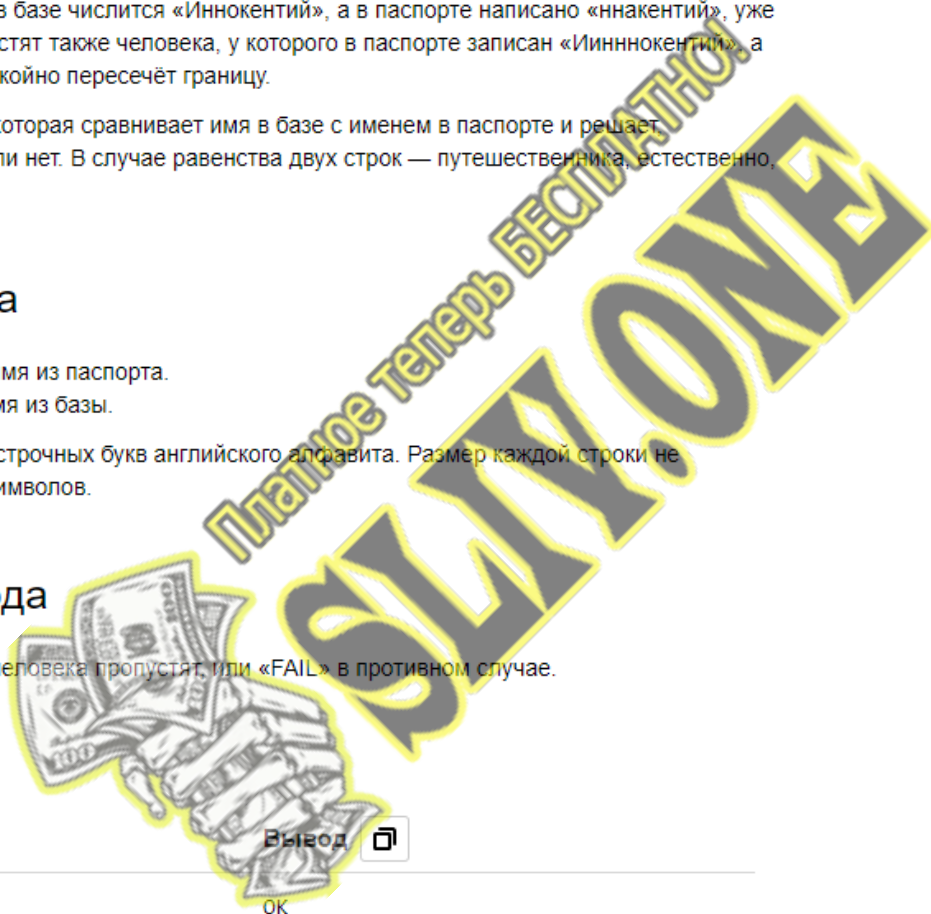
Ввод

mata
para

Вывод

FAIL

- [A. Разворот строки](#)
- [B. Пограничный контроль](#)
- [C. Самый длинный палиндром 2](#)
- [D. Общий префикс](#)
- [E. Вставка строк](#)
- [F. Частое слово](#)
- [G. Поиск со сдвигом](#)
- [H. Глобальная замена](#)
- [I. Повтор](#)
- [J. Случай верблюда](#)
- [K. Сравнить две строки](#)
- [L. Подсчёт префикс-функции](#)



Яндекс.Практикум, Алгоритмы: Спринт 8. Финальные задачи

[Положение участников](#) [Задачи](#) [Посылки](#)

В. Шпаргалка

[A. Packed Prefix](#)

В. Шпаргалка

Ограничение времени	0.7 секунд
Ограничение памяти	256Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Вася готовится к экзамену по алгоритмам и на всякий случай пишет шпаргалки. Чтобы уместить на них как можно больше информации, он не разделяет слова пробелами. В итоге получается одна очень длинная строка. Чтобы на самом экзамене из-за нервов не запутаться в прочитанном, он просит вас написать программу, которая по этой длинной строке и набору допустимых слов определит, можно ли разбить текст на отдельные слова из набора.

Более формально: дан текст T и набор строк s_1, \dots, s_n . Надо определить, представим ли T как $s_{k_1}s_{k_2}\dots s_{k_r}$, где k_i — индексы строк. Индексы могут повторяться. Строка s_i может встречаться в разбиении текста T произвольное число раз. Можно использовать не все строки для разбиения. Строки могут идти в любом порядке.

Формат ввода

В первой строке дан текст T , который надо разбить на слова. Длина T не превосходит 10^5 . Текст состоит из строчных букв английского алфавита.

Во второй строке записано число допустимых к использованию слов $1 \leq n \leq 100$.

В последующих n строках даны сами слова, состоящие из маленьких латинских букв. Длина каждого слова не превосходит 100 .

Формат вывода

Выведите «YES», если текст можно разбить на слова из данного словаря, или «NO» в ином случае.

Пример 1

Ввод

Вывод

```
examwillpasstheexam
5
will
pass
the
exam
i
```

YES

Пример 2

Ввод

Вывод

```
abacaba
2
abac
caba
```

NO

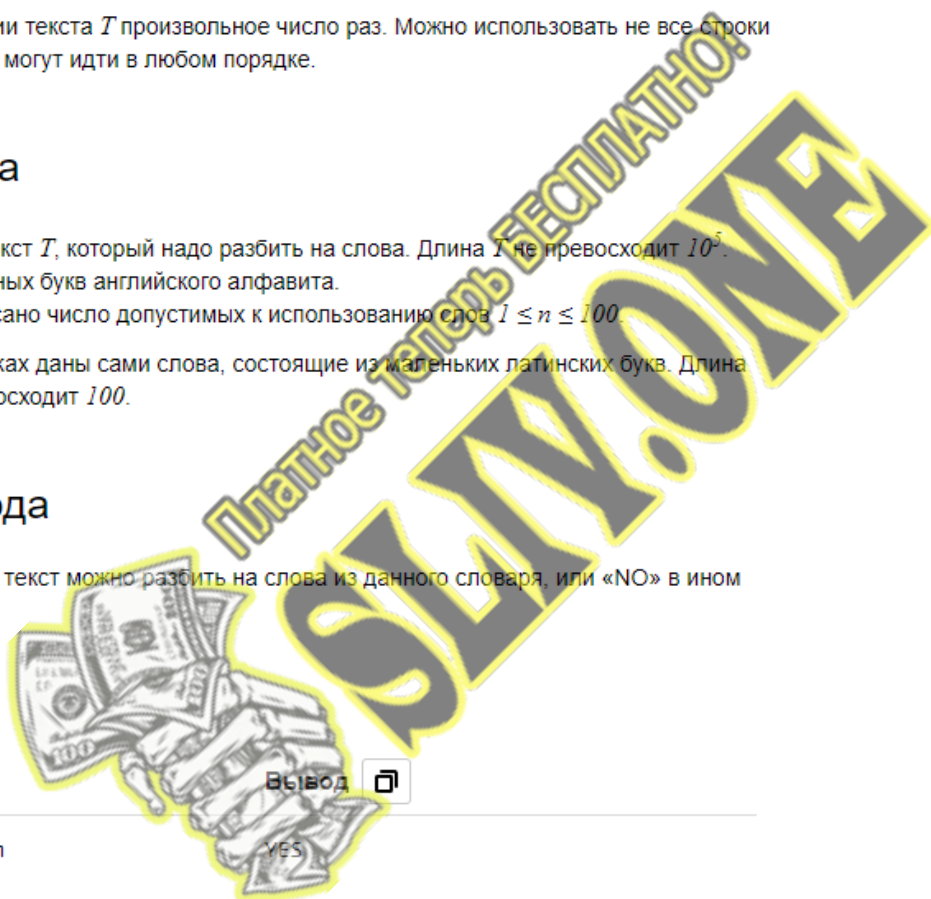
Пример 3

Ввод

Вывод

```
abacaba
3
abac
caba
aba
```

YES



С. Самый длинный палиндром 2

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Палиндром — это строка, которая одинаково читается как слева направо, так и справа налево.

Из данной строки s путём удаления и перестановки букв надо получить палиндром максимальной длины. Среди всех таких палиндромов надо получить лексикографически минимальный. Количество удалений и перестановок символов может быть любым.

Формат ввода

В единственной строке дана строка s . Длина строки $|s| \leq 10^5$. Строка состоит из строчных букв английского алфавита.

Формат вывода

Выведите полученный палиндром. Заметьте, что ответ определяется однозначно.

Пример 1

Ввод

aaaabb

Вывод

aabbaa

Пример 2

Ввод

rabcd

Вывод

a

Пример 3

Ввод

aaabbb

Вывод

ababa

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции



D. Общий префикс

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	1 секунда	128Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Node.js 14.15.5	2 секунды	128Mb		
Python 3.7.3	5 секунд	64Mb		
Mono C# 5.2.0	2 секунды	256Mb		
Oracle Java 8	2 секунды	256Mb		
OpenJDK Java 11	2 секунды	256Mb		
Node JS 8.16	2 секунды	128Mb		

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции

Найдите наибольший по длине общий префикс нескольких строк.

Формат ввода

В первой строке дано число n ($1 \leq n \leq 10^5$). Затем по одной на строке даны n строк, каждая не превышает 10^5 в длину. Суммарная длина всех строк не превосходит 10^7 .

Формат вывода

Выведите единственное число — длину наибольшего префикса всех данных строк.

Пример 1

Ввод

```
3
abacaba
abudabi
abcdefg
```

Вывод

```
2
```

Пример 2

Ввод

```
2
tutu
kukuku
```

Вывод

```
0
```

Пример 3

Ввод

```
3
qwe
qwerty
qwerpy
```

Вывод

```
3
```



Е. Вставка строк

	Все языки	GNU c++17 7.3
Ограничение времени	1 секунда	0.2 секунды
Ограничение памяти	64Mb	64Mb
Ввод	стандартный ввод или input.txt	
Вывод	стандартный вывод или output.txt	

У Риты была строка s , Гоша подарил ей на 8 марта ещё n других строк t_i , $1 \leq i \leq n$. Теперь Рита думает, куда их лучше поставить. Один из вариантов — расположить подаренные строки внутри имеющейся строки s , поставив строку t_i сразу после символа строки s с номером k_i (в частности, если $k_i=0$, то строка вставляется в самое начало s).

Помогите Рите и определите, какая строка получится после вставки в s всех подаренных Гошей строк.

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции

Формат ввода

В первой строке дана строка s . Строка состоит из строчных букв английского алфавита, не бывает пустой и её длина не превышает 10^5 символов.

Во второй строке записано количество подаренных строк — натуральное число n , $1 \leq n \leq 10^5$.

В каждой из следующих n строк через пробел записаны пары t_i и k_i . Строка t_i состоит из маленьких латинских букв и не бывает пустой, k_i — целое число, лежащее в диапазоне от 0 до $|s|$. Все числа k_i уникальны. Гарантируется, что суммарная длина всех строк t_i не превосходит 10^5 .

Формат вывода

Выведите получившуюся в результате вставок строку.

Пример 1

Ввод

```
abacaba
3
queue 2
deque 0
stack 7
```

Вывод

```
dequeabqueueacabastack
```

Пример 2

Ввод

```
kukareku
2
p 1
q 2
```

Вывод

```
кpуqкareku
```

Ф. Частое слово

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.6 секунд	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Mono C# 5.2.0	0.6 секунд	128Mb		
Oracle Java 8	0.6 секунд	128Mb		
OpenJDK Java 11	0.6 секунд	128Mb		

Дан набор слов, слова могут повторяться. Среди них надо найти самое частое слово. Если таких слов несколько, то выведите лексикографически наименьшее.

Формат ввода

В первой строке дано число n ($1 \leq n \leq 10^3$) — количество строк. В следующих n строках даны слова, состоящие из строчных букв английского алфавита. Слово не бывает пустым.

Суммарная длина слов не превосходит 10^7 . Длина одного слова не превосходит 10^4 .

Формат вывода

Выведите единственную строку — наиболее частое слово, лексикографически минимальное, если самых частых слов несколько.

Пример 1

Ввод

```
5
caba
aba
caba
abac
aba
```

Вывод

```
aba
```

Пример 2

Ввод

```
3
b
bc
bcd
```

Вывод

```
b
```

Пример 3

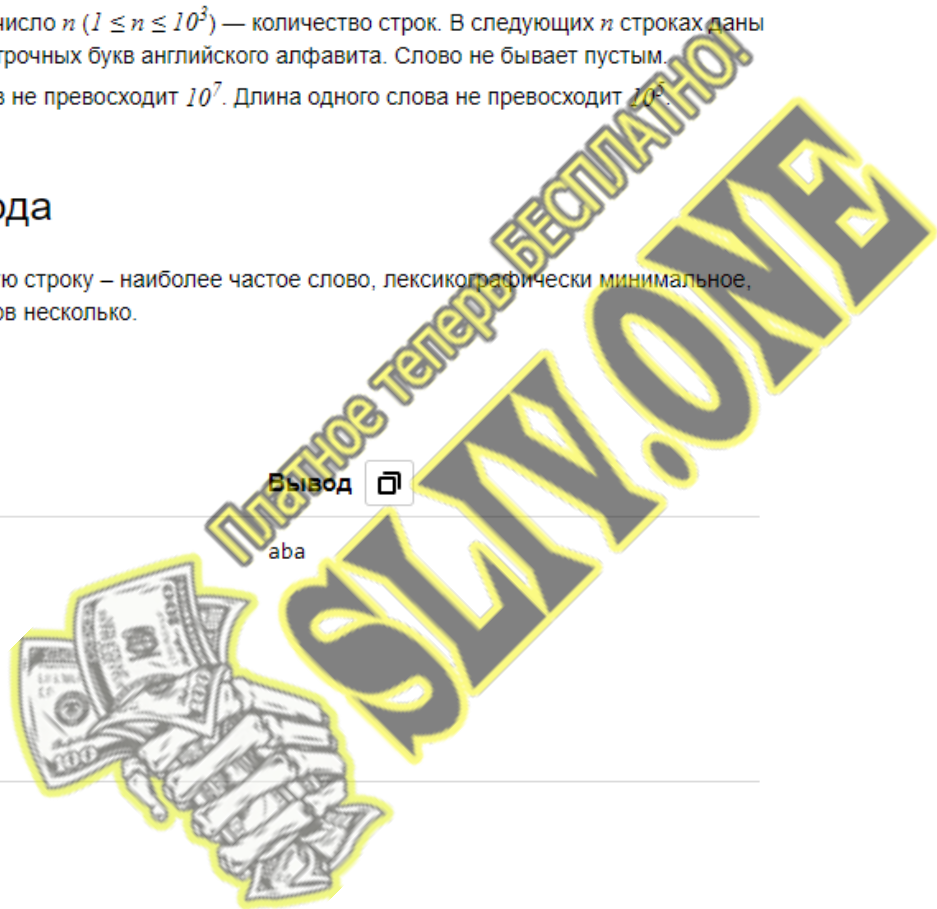
Ввод

```
10
ciwlaxtnhhrnenw
ciwnvsuni
ciwaxeujmsmvpojqqkxk
ciwnvsuni
ciwnvsuni
ciwuxlkecnofovq
ciwuxlkecnofovq
ciwodramivid
ciwlaxtnhhrnenw
ciwnvsuni
```

Вывод

```
ciwnvsuni
```

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции



G. Поиск со сдвигом

	Все языки	Python 3.7.3	GNU c++17 7.3
Ограничение времени	1.5 секунд	3 секунды	0.15 секунд
Ограничение памяти	64Mb	64Mb	64Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

Гоша измерял температуру воздуха n дней подряд. В результате у него получился некоторый временной ряд. Теперь он хочет посмотреть, как часто встречается некоторый шаблон в получившейся последовательности. Однако температура — вещь относительная, поэтому Гоша решил, что при поиске шаблона длины m (a_1, a_2, \dots, a_m) стоит также рассматривать сдвинутые на константу вхождения. Это значит, что если для некоторого числа c в исходной последовательности нашёлся участок вида $(a_1 + c, a_2 + c, \dots, a_m + c)$, то он тоже считается вхождением шаблона (a_1, a_2, \dots, a_m) .

По заданной последовательности измерений X и шаблону $A=(a_1, a_2, \dots, a_m)$ определите все вхождения A в X , допускающие сдвиг на константу.

Подсказка: если вы пишете на питоне и сталкиваетесь с TL, то попробуйте заменить какие-то из циклов операциями со срезами.

Формат ввода

В первой строке дано количество сделанных измерений n — натуральное число, не превышающее 10^4 . Во второй строке через пробел записаны n целых чисел x_i , $0 \leq x_i \leq 10^3$ — результаты измерений. В третьей строке дано натуральное число m — длина искомого шаблона, $1 \leq m \leq n$. В четвёртой строке даны m целых чисел a_i — элементы шаблона, $0 \leq a_i \leq 10^3$.

Формат вывода

Выведите через пробел в порядке возрастания все позиции, на которых начинаются вхождения шаблона A в последовательность X . Нумерация позиций начинается с единицы.

Пример 1

Ввод

```
9
3 9 1 2 5 10 9 1 7
2
4 10
```

Вывод

```
1 8
```

Пример 2

Ввод

```
5
1 2 3 4 5
3
10 11 12
```

Вывод

```
1 2 3
```

- [A. Разворот строки](#)
- [B. Пограничный контроль](#)
- [C. Самый длинный палиндром 2](#)
- [D. Общий префикс](#)
- [E. Вставка строк](#)
- [F. Частое слово](#)
- [G. Поиск со сдвигом](#)
- [H. Глобальная замена](#)
- [I. Повтор](#)
- [J. Случай верблюда](#)
- [K. Сравнить две строки](#)
- [L. Подсчёт префикс-функции](#)

Н. Глобальная замена

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Напишите программу, которая будет заменять в тексте все вхождения строки s на строку t . Гарантируется, что никакие два вхождения шаблона s не пересекаются друг с другом.

Формат ввода

В первой строке дан текст — это строка из строчных букв английского алфавита, длина которой не превышает 10^6 .

Во второй строке записан шаблон s , вхождения которого будут заменены.


В третьей строке дана строка t , которая будет заменять вхождения.

Обе строки s и t состоят из строчных букв английского алфавита, длина каждой строки не превосходит 10^5 . Размер итоговой строки не превосходит $2 \cdot 10^6$.


Формат вывода

В единственной строке выведите результат всех замен — текст, в котором все вхождения s заменены на t .

Пример 1


Ввод 

```
pingpong  
ng  
mpi
```


Вывод 

```
pimpipompi
```

Пример 2

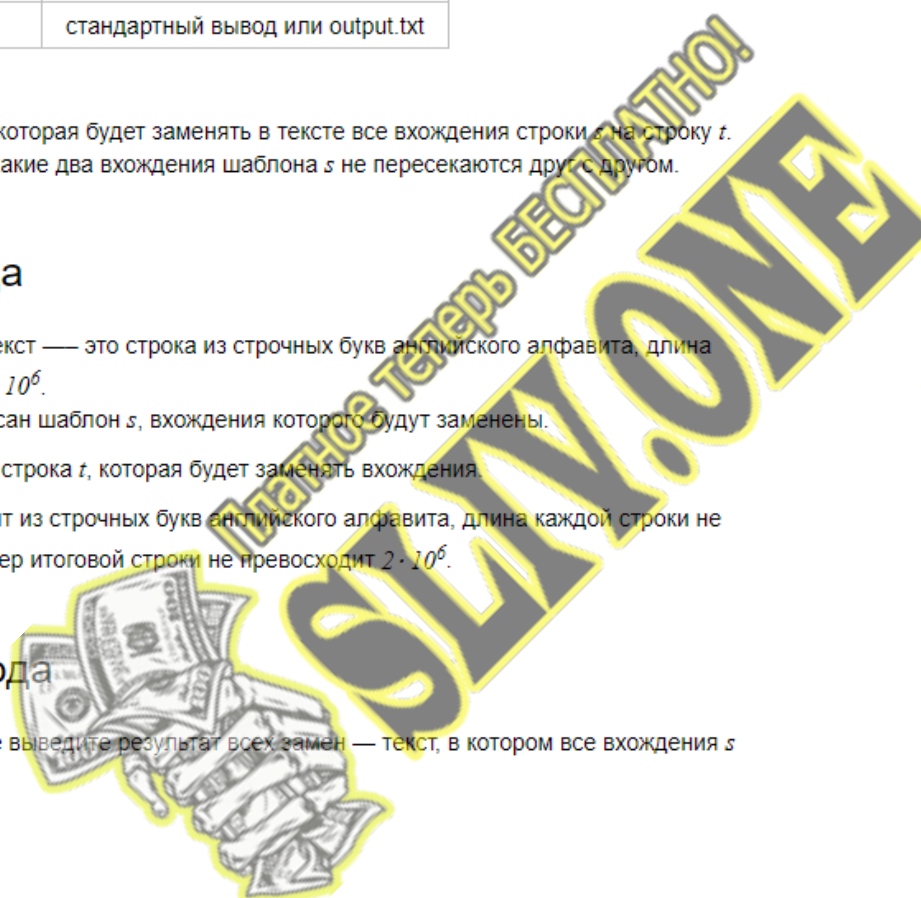
Ввод 

```
aaa  
a  
ab
```

Вывод 

```
ababab
```

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- N. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции



I. Повтор

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	0.3 секунды	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	0.6 секунд	64Mb		
Mono C# 5.2.0	0.4 секунды	128Mb		
Oracle Java 8	0.4 секунды	128Mb		
OpenJDK Java 11	0.4 секунды	128Mb		

Будем говорить, что строка s является повтором длины k , если существует такая строка t , что $s = t * k$, где под умножением подразумевается конкатенация k экземпляров строки t один за другим.

Например, строка $abababab$ является повтором строки $abab$ длины 2, а также повторением строки ab длины 4. Тогда имеет смысл говорить о **наибольшем повторе**. Строка является наибольшим повтором длины k , если она является повтором некоторой строки длины k и если не существует такой строки t , что s — повтор t длины $m > k$. Например, строка $aaaa$ является наибольшим повтором длины 4.

Вам дана строка, которая является наибольшим повтором длины x . Найдите x .

Заметим, что ответ всегда равен хотя бы единице, так как строка является повтором самой себя.

Формат ввода

В единственной строке дана строка, состоящая из строчных букв английского алфавита и не превышающая в длину 10^6 . Строка не бывает пустой.

Формат вывода

Выведите единственное число — x , длину наибольшего повтора.

Пример 1

Ввод

zzzzzz

Вывод

6

Пример 2

Ввод

abacaba

Вывод

1

Пример 3

Ввод

abababab

Вывод

4

- [A. Разворот строки](#)
- [B. Пограничный контроль](#)
- [C. Самый длинный палиндром 2](#)
- [D. Общий префикс](#)
- [E. Вставка строк](#)
- [F. Частое слово](#)
- [G. Поиск со сдвигом](#)
- [H. Глобальная замена](#)
- [I. Повтор](#)
- [J. Случай верблюда](#)
- [K. Сравнить две строки](#)
- [L. Подсчёт префикс-функции](#)

J. Случай верблюда

Язык	Ограничение времени	Ограничение памяти	Ввод	Вывод
Все языки	1 секунда	64Mb	стандартный ввод или input.txt	стандартный вывод или output.txt
Python 3.7.3	5 секунд	64Mb		
Mono C# 5.2.0	3 секунды	256Mb		
Oracle Java 8	3 секунды	256Mb		
OpenJDK Java 11	3 секунды	256Mb		

В некоторых IDE поддерживается навигация по файлам через их сокращённые названия. Если в языке принято называть классы CamelCase'ом (как в Java, например), то по заглавным буквам названия можно быстро найти нужный класс. Например, если название класса «MyFavouriteConfigurableScannerFactory», то его можно найти по строке «MFCSF». Но если в проекте есть класс «theMultiFunctionalCommaSeparatedFile», то он тоже будет подходить под этот паттерн, и при поиске надо будет выбрать между этими двумя вариантами.

Вам дан набор строк в CamelCase. Далее будут поступать запросы в виде строк-паттернов из прописных букв английского алфавита. Вам надо находить такие строки среди исходных, которые удовлетворяют заданному шаблону, и выводить их в лексикографическом порядке.

Также в паттерне может быть только несколько первых заглавных букв. Например, если бы в указанном выше примере был бы паттерн «MFCS», то существующие две строки подошли бы под него, а также подходил бы, например, «MamaFicusCodingSouthWestNorth». А вот «MamaCodingSouthWestNorth» — уже нет.

Формат ввода

В первой строке записано число — количество названий классов в исходном наборе n ($1 \leq n \leq 10^5$). Все названия состоят из строчных и прописных букв английского алфавита.

В следующих n строках даны сами названия по одному в строке. Суммарная длина этих строк не превосходит 10^7 .

Затем дано количество запросов m .

В следующих m строках даны сами запросы. Каждый запрос — это шаблон, строка из прописных букв английского алфавита, в длину не превышающая 10^5 . Шаблон может быть пустым. Заметьте: шаблону из нуля прописных букв удовлетворяет любое название.

Формат вывода

Для каждого отдельного запроса (в порядке их поступления) выведите в лексикографическом порядке все строки, которые подходят под данный шаблон. Если какие-то строки одинаковые, то выведите все экземпляры. Если ни одна из строк не подходит под шаблон, то выведите для данного запроса пустую строку.

Пример 1

Ввод

```
3
МамаMilaRamu
МамаMia
MonAmi
2
MM
MA
```

Вывод

```
МамаMia
МамаMilaRamu
MonAmi
```

Пример 2

Ввод

```
2
AlphaBetaGamma
AbcdBcdGggg
2
ABGG
ABG
```

Вывод

```
AbcdBcdGggg
AlphaBetaGamma
```

Пример 3

Ввод

```
5
WudHnagkbhfwrbcI
WCUkvoxboxufsdap
jdrxomezrpuhbgI
ZcGHdrPplfoldemu
cylbtqwuxhiveznc
3
WGHV
NKVDT
ZGHU
```

Вывод

- [A. Разворот строки](#)
- [B. Пограничный контроль](#)
- [C. Самый длинный палиндром 2](#)
- [D. Общий префикс](#)
- [E. Вставка строк](#)
- [F. Частое слово](#)
- [G. Поиск со сдвигом](#)
- [H. Глобальная замена](#)
- [I. Повтор](#)
- [J. Случай верблюда](#)
- [K. Сравнить две строки](#)
- [L. Подсчёт префикс-функции](#)

К. Сравнить две строки

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Алла придумала новый способ сравнивать две строки: чтобы сравнить строки a и b , в них надо оставить только те буквы, которые в английском алфавите стоят на четных позициях. Затем полученные строки сравниваются по обычным правилам. Помогите Алле реализовать новое сравнение строк.

Формат ввода

На вход подаются строки a и b по одной в строке. Обе строки состоят из маленьких латинских букв, не бывают пустыми и не превосходят 10^5 символов в длину.

Формат вывода

Выведите -1 , если $a < b$, 0 , если $a = b$ и 1 , если $a > b$.

Пример 1

Ввод

```
gggggbbb  
bbef
```

Вывод

```
-1
```

Пример 2

Ввод

```
z  
aaaaaaa
```

Вывод

```
1
```

Пример 3

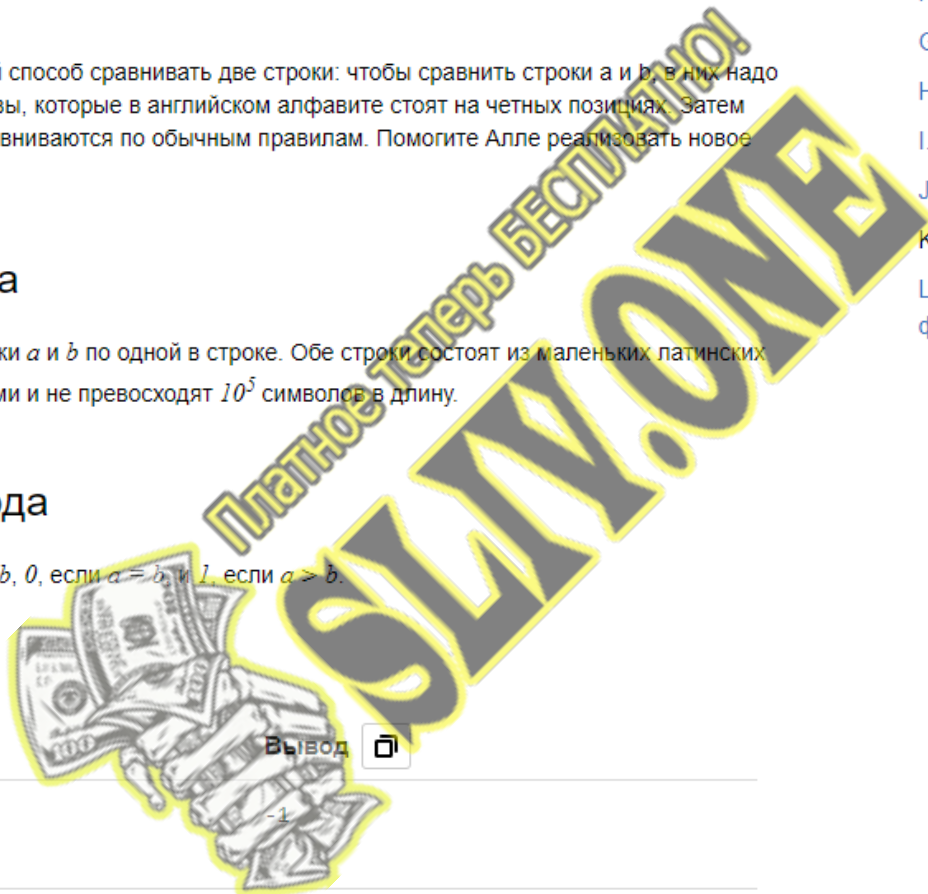
Ввод

```
ccccz  
aaaaaz
```

Вывод

```
0
```

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции



L. Подсчёт префикс-функции

	Все языки	Python 3.7.3	GNU c++17 7.3
Ограничение времени	1 секунда	1.6 секунд	0.4 секунды
Ограничение памяти	128Mb	128Mb	64Mb
Ввод	стандартный ввод или input.txt		
Вывод	стандартный вывод или output.txt		

В этой задаче вам необходимо посчитать префикс-функцию для заданной строки.

Формат ввода

На вход подаётся строка, состоящая из строчных латинских букв. Длина строки не превосходит 10^6 .

Формат вывода

Если длина входной строки L , то выведите через пробел L целых неотрицательных чисел — массив значений префикс-функции исходной строки.

Пример 1

Ввод

abracadabra

Вывод

0 0 0 1 0 1 0 1 2 3 4

Пример 2

Ввод

xxzzxxz

Вывод

0 1 0 0 1 2 3

Пример 3

Ввод

aaaaa

Вывод

0 1 2 3 4

- A. Разворот строки
- B. Пограничный контроль
- C. Самый длинный палиндром 2
- D. Общий префикс
- E. Вставка строк
- F. Частое слово
- G. Поиск со сдвигом
- H. Глобальная замена
- I. Повтор
- J. Случай верблюда
- K. Сравнить две строки
- L. Подсчёт префикс-функции

