

Строки.

Почитать про строки можно здесь: справочник, стр. 15 (глава 3) и стр. 37 (глава 8).

Во всех задачах запрещается использование констант, обозначающих порядковые номера любых символов в таблице ASCII.

Для изменения строки лучше пользоваться следующим приёмом. Сперва преобразовать строку в массив отдельных символов: `x = list(s)`, сделать все необходимые изменения (массивы допускают изменения своих элементов в отличие от строк), затем вывести получившуюся строку:

```
print(''.join(x)).
```

A. Палиндром

Строка называется *палиндромом*, если она читается слева направо и справа налево одинаково.

Программа должна вывести слово YES, если введённое слово — палиндром, и слово NO, если оно не является палиндромом.

Решите эту задачу, используя не более $N//2$ операций сравнения символов и не используя сравнений строк и их срезов.

| Input | Output |
|-------|--------|
| kazak | YES |

B. Количество слов

Дана строка, содержащая произвольные символы. Определите количество слов в этой строке.

Слово — это несколько подряд идущих букв латинского алфавита (как заглавных, так и строчных), ограниченных слева и справа символами-не-буквами или началом/концом строки.

В примере входных данных пробелы обозначены символом ~.

| Input | Output |
|---|--------|
| Yesterday,~all~my~troubles~seemed~so~far~away | 8 |
| hmm,wrong~spaces~~here | 4 |

C. Самое длинное слово

Напишите программу, которая выводит самое длинное слово переданной ей символьной строки. Слово — это последовательность символов, отличных от пробела, ограниченная пробелами или концами строки.

Программа должна вывести в первой строке самое длинное слово переданной ей строки, а во второй — длину этого слова. Если слов максимальной длины несколько — вывести первое встретившееся слово максимальной длины.

| Input | Output |
|--------------------|--------------|
| abra cadabra fibra | cadabra 7 |

D. Замена регистра - I

Дана строка. Напечатать строку, в которой вместо строчных букв исходной строки будут соответствующие прописные и наоборот.

| Input | Output |
|-------|--------|
| hELLO | Hello |

E. Красные и синие - I

Дана цепочка, состоящая из синих (B) и красных (R) точек, всего не более 500000 точек. Требуется выяснить, какое минимальное количество синих точек можно удалить так, чтобы сначала шли только синие, а потом — только красные.

В этой задаче запрещается использовать вспомогательные строки (массивы, срезы) размера, зависящего от размера входных данных, а также сравнивать строки, а не отдельные символы.

Нужно вывести полученную цепочку, в которой сначала идут только синие точки, а потом — только красные. Во второй строке нужно вывести количество удалённых синих точек.

Здесь полезно не моделировать процесс удаления символов для получения правильного результата, а подсчитать количество букв R и B в получающейся строке. Потом использовать умножение строки на число и сложение строк.

| Input | Output |
|-----------|--------------|
| BBBRBRBRR | BBBRRRR 2 |

F. Слова наоборот

На вход программе подаётся строка, содержащая слова, разделенные пробелами (можно считать, что строка содержит только строчные буквы и пробелы и есть как минимум одно слово).

Программа должна напечатать строку, содержащую те же слова в обратном порядке, которые разделены *одним* пробелом (сами слова не меняются, меняется их порядок).

Напечатанная строка не должна начинаться с пробела или заканчиваться им.

В примере входных данных пробелы обозначены символом ~.

| Input | Output |
|-------------------------------|----------------|
| ~~~~abcd~~~~efgh~~~~~prst~~~~ | prst efgh abcd |

G. Количество чисел

Дана строка, содержащая произвольные символы. Посчитать количество *натуральных чисел*, записанных в этой строке.

Натуральное число — последовательность цифр, начинающаяся не с нуля и не являющаяся частью другой последовательности, образующей натуральное число.

Например:

для входной строки `abc123 2023 000340004` программа должна вывести число 3

Комментарий к первому примеру: строки '0004', '23', '40004' не являются записью натурального числа в смысле данного выше определения, т.к. содержатся как подстроки в других натуральных числах, соответственно '340004', '2023'.

| Input | Output |
|-----------------------|--------|
| abc123 2023 000340004 | 3 |
| Input | Output |
| 2 0000 0 00 | 1 |

H. Шифр Цезаря

Юлий Цезарь использовал следующий способ шифрования текста: каждая буква заменялась на следующую по алфавиту через K позиций по кругу. То есть, например, при $K = 2$ буква В заменялась на D, буква X заменялась на Z, а буква Y заменялась на A.

Вам задана строка, состоящая только из заглавных букв латинского алфавита — результат шифровки, и число K ($0 \leq K \leq 10^6$), использованное при шифровании. Необходимо по этим данным определить исходный текст.

| Input | Output |
|-----------|--------|
| XPSE 1 | WORD |

I. IP-адрес

IP-адрес это четырёхбайтовый код, который принято записывать в виде четырех десятичных чисел, разделенных точками. Каждое из чисел может принимать значения от 0 до 255. Вот примеры правильных IP-адресов:

127.0.0.0

192.168.0.1

255.0.255.255

Напишите функцию, которая будет возвращать `True`, если переданная строка является правильным IP-адресом, и `False` в противном случае.

На вход программе подаётся произвольная строка. Программа должна вывести строку `YES`, если это правильный IP-адрес и `NO` в противном случае.

| Input | Output |
|-----------|--------|
| 127.0.0.1 | YES |

J. Калькулятор

Напишите программу, которая вычисляет арифметическое выражение, введённое в виде символьной строки. Выражение содержит только целые числа и знаки сложения и вычитания.

Функцию `eval` использовать нельзя.

| Input | Output |
|---------------|--------|
| 1+12-54+68-17 | 10 |

К. Распаковка строки

Будем рассматривать только строчки, состоящие из заглавных латинских букв. Например, рассмотрим строку AAAABCCCCDDDD. Длина этой строки равна 14. Поскольку строка состоит только из латинских букв, повторяющиеся символы могут быть удалены и заменены числами, определяющими количество повторений. Таким образом, данная строка может быть представлена как 4A5C4D. Длина такой строки 7. Описанный метод мы назовём упаковкой строки.

Напишите программу, которая берёт упакованную строчку и восстанавливает по ней исходную строку.

Входной файл содержит одну упакованную строку. В строке могут встречаться только конструкции вида nA , где n — количество повторений символа (целое число от 2 до 99), а A — заглавная латинская буква, либо конструкции вида A , то есть символ без числа, определяющего количество повторений. Максимальная длина строки не превышает 80.

В выходной файл выведите восстановленную строку. При этом строка должна быть разбита на строчки длиной ровно по 40 символов (за исключением последней, которая может содержать меньше 40 символов).

| Input | Output |
|-------|--------|
| AB2C | ABCC |

Л. Входит ли одна строка в другую

Даны две строки: T и P . Найти количество вхождений строки P в строку T , т.е. количество таких индексов строки T , начиная с которых строка P входит в строку T .

Решите эту задачу без использования дополнительных строк и срезов, а также без сравнения срезов (разрешается сравнивать только отдельные символы).

На вход программе подаётся две строки: в первой строка T , во второй строка P . Программа должна вывести одно целое неотрицательное число.

| Input | Output |
|---------------------------|--------|
| AAABAABVBAABVBAVBA AAB | 3 |

| Input | Output |
|-------------|--------|
| AAAAA AA | 4 |

М. Форматирование строки

Дана текстовая строка, содержащая буквы латинского алфавита, пробелы, запятые и точки. Отформатируйте этот текст по следующим правилам:

- В начале и конце строки не должно быть пробелов.
- Все слова разделяются ровно одним пробелом.
- Точки и запятые пишутся слитно с предыдущим словом, после знака препинания ставится пробел.

Выведите полученную строку. После неё обязательно следовать символ перевода строки.

Гарантируется, что никакие два знака препинания не идут подряд.

В примере входных и выходных данных пробелы обозначены символом `~`.

| Input | Output |
|------------------------|---------------|
| ~~~~~Hello~,~world.~~~ | Hello,~world. |

Н. Из десятичной в римскую

Напишите программу, которая выводит запись заданного десятичного числа, не превосходящего 3999, в римской системе счисления. Подробнее о правилах перевода [здесь \(link\)](#).

| Input | Output |
|-------|----------|
| 1234 | MCCXXXIV |

О. Палиндромы без учёта регистра

Дано слово, состоящее только из заглавных и строчных латинских букв. Проверьте, является ли это слово палиндромом, если считать заглавные и строчные буквы не различающимися. Выведите слово YES, если слово является таким палиндромом и слово NO, если не является.

Решение оформите в виде функции `IsPalindrome(S)`, возвращающей значение типа `bool`. При решении этой задачи нельзя изменять входную строку, пользоваться вспомогательными массивами или строками.

| Input | Output |
|-------|--------|
| Radar | YES |

P. *Палиндромы без учёта пробелов*

Дана строка, состоящая из строчных латинских букв и пробелов. Проверьте, является ли она палиндромом без учета пробелов (например, "аргентина манит негра").

В этой задаче запрещается изменять входную строку, использовать вспомогательные строки, срезы, а также сравнивать строки, а не отдельные символы.

| Input | Output |
|-------|--------|
| ab a | YES |

Q. *Палиндромы без учёта гласных букв*

Возьмем произвольное слово и сделаем с ним следующую операцию: поменяем местами его первую согласную букву с последней согласной буквой, вторую согласную букву с предпоследней согласной буквой и т.д. Если после этой операции мы вновь получим исходное слово, то будем называть такое слово негласным палиндромом.

Например, слова *sos, rare, rotor, gong, karaoke* являются негласными палиндромами.

Вам требуется написать программу, которая по данному слову определяет, является ли оно негласным палиндромом.

Перечень согласных букв английского алфавита вам предлагается составить самостоятельно.

В этой задаче запрещается изменять входную строку, использовать вспомогательные строки, срезы, а также сравнивать строки, а не отдельные символы.

| Input | Output |
|---------|--------|
| tennete | YES |

R. *Магическая последовательность*

Даны последовательности: 1, 11, 21, 1211, 111221, 312211, 13112221, 1113213211, ...

Выпишите k -ю последовательность.

| Input | Output |
|-------|--------|
| 4 | 1211 |

S. *Majority*

Известно, что в строке один из символов встречается чаще остальных вместе взятых. Вывести этот символ, сделав один проход по строке. В качестве дополнительных переменных разрешается использовать только строки длины 1 и целые числа. Исходную строку изменять нельзя.

На вход программе подаётся строка, содержащая не более 10^6 символов.

| Input | Output |
|------------|--------|
| abcabcaaaa | a |

T. *Замена регистра - II*

Дана строка, содержащая произвольные символы. Измените регистр символов, являющихся буквами латинского алфавита в этой строке так, чтобы первая буква каждого слова была заглавной, а остальные буквы — строчными.

Слово — это последовательность подряд идущих букв, ограниченная слева и справа либо концом исходной строки, либо символами не буквами.

| Input | Output |
|--------------------|--------------------|
| this is an example | This Is An Example |

U. *Красные и синие - II*

Дана цепочка, состоящая из синих (B) и красных (R) точек, состоящая не более, чем из $5 \cdot 10^5$ символов. Нужно удалить наименьшее количество красных точек так, чтобы сначала шли только синие, а потом — только красные.

В этой задаче запрещается использовать вспомогательные строки, срезы, а также сравнивать строки, а не отдельные символы.

Нужно вывести полученную цепочку, в которой сначала идут только синие точки, а потом — только красные. Во второй строке нужно вывести количество удалённых красных точек.

| Input | Output |
|-------------------|-----------------|
| BBBRBRBRRRRBBBRRR | BBBBBBBRRR 5 |

V. Красные и синие - III

Дана цепочка, состоящая из синих (B) и красных (R) точек, содержащая не более $5 \cdot 10^5$ символов. Нужно удалить наименьшее одинаковое количество синих и красных точек так, чтобы сначала шли только синие, а потом — только красные.

В этой задаче запрещается использовать вспомогательные строки, срезы, а также сравнивать строки, а не отдельные символы.

Нужно вывести полученную цепочку, в которой сначала идут только синие точки, а потом — только красные. Во второй строке нужно вывести количество удалённых (синих и красных) точек.

| Input | Output |
|----------------|-------------------|
| BBBBBBRRRRRRRR | BBBBBBRRRRRR 4 |

W. Метод бутерброда

Секретное агентство «Super-Secret-но» решило для шифрования переписки своих сотрудников использовать «метод бутерброда».

Сначала буквы слова нумеруются в таком порядке: первая буква получает номер 1, последняя буква — номер 2, вторая — номер 3, предпоследняя — номер 4, потом третья ... и так для всех букв. Затем все буквы записываются в шифр в порядке своих номеров. В конец зашифрованного слова добавляется знак «дизел» (#), который нельзя использовать в сообщениях.

Например, слово `sandwich` зашифруется в `shacnidw#`.

Вводится слово, зашифрованное методом бутерброда. Длина слова не превышает 20 букв.

Выведите расшифрованное слово. После слова обязательно вывести перенос строки.

| Input | Output |
|--------------|-------------|
| Aabrrbaacda# | Abracadabra |

X. Маска файла

Миша готовится к ЕГЭ по информатике. Сейчас он изучает задачу A4, в которой описывается работа с масками файлов.

Для групповых операций с файлами используются маски имён файлов. Маска представляет собой последовательность букв, цифр и прочих допустимых в именах файлов символов, в которой также могут встречаться следующие символы.

Символ «?» (вопросительный знак) означает ровно один произвольный символ.

Символ «*» (звёздочка) означает любую последовательность символов произвольной длины, в том числе «*» может задавать и пустую последовательность.

Напишите программу, которая для каждого имени файла определит, подходит ли оно под заданную маску. Гарантируется, что в маске файла присутствует не более одного символа «*».

| Input | Output |
|----------------------|--------|
| ?or*.d?? fort.doc | YES |

Y. Сделать палиндромом

Дано слово, состоящее только из строчных латинских букв. Определите, какое наименьшее число букв нужно дописать к этому слову справа так, чтобы оно стало палиндромом.

| Input | Output |
|-------|--------|
| abcd | 3 |

Z. Следующий палиндром

Рассмотрим все натуральные числа, запись которых в десятичной системе счисления является палиндромом (при этом запись не начинается с нуля). Например, числа 121 и 1331 являются палиндромами, а число 123 — нет. По данному натуральному числу N определите следующее за ним натуральное число (то есть наименьшее число, которое превосходит N), являющееся палиндромом.

Программа получает на вход одно натуральное число N , состоящее не более чем из 200 цифр.

Программа должна вывести наименьшее натуральное число, которое больше N и является палиндромом.

| Input | Output |
|-------|--------|
| 4321 | 4334 |