

Строки.

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

Справочник на английском, описывающий стандартные методы работы со строками: документация по языку Python.

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)).
```

Термин ACGT-строка означает строку произвольной длины, состоящую только из заглавных символов A, C, G и T.

A. Количество символов

Дана ACGT-строка. Подсчитать и вывести количество символов A, C, G и T.

Input	Output
ACGTT	1 1 1 2

B. Извлечь цифры

Дана произвольная строка. Извлеките из этой строки все символы, являющиеся цифрами и составьте из них новую строку.

Программа получает на вход одну строку.

Требуется вывести новую строку, содержащую только цифры данной строки в том же порядке, в котором они в ней следовали.

Input	Output
2+2=4	224

C. Значение выражения - 1

Дана строка, состоящая из N цифр (т.е. однозначных чисел), между которыми стоит $N - 1$ знак операции, каждый из которых может быть либо +, либо -. Вычислите значение данного выражения.

Input	Output
1+2-3	0

D. Значение выражения - 2

Дано выражение одно из следующих видов: $A+B$, $A-B$ или $A*B$, где A и B — целые числа от 0 до 10^9 . Определите значение этого выражения.

Input	Output
100-101	-1
2*2	4

E. Капитан Флинт

Капитан Флинт зарыл клад на Острове сокровищ. Он оставил описание, как найти клад. Описание состоит из строк вида: North 5, где первое слово — одно из North, South, East, West, а второе число — количество шагов, необходимое пройти в этом направлении.

Напишите программу, которая по описанию пути к кладу определяет точные координаты клада, считая, что начало координат находится в начале пути, ось OX направлена на восток, ось OY — на север.

Программа получает на вход последовательность строк указанного вида, завершающуюся строкой со словом **Treasure!**

Программа должна вывести два целых числа: координаты клада (сначала абсциссу, потом ординату).

Input	Output
North 5 East 3 South 1 Treasure!	3 4

Ф. *Наименьший триплет - 1*

Дана строка, содержащая не менее трёх символов. *Триплетом* называется подстрока из трёх соседних символов.

Вывести наименьший в лексикографическом порядке триплет данной строки. Рассматриваются **ВСЕ** триплеты, начинающиеся с индексов 0, 1 и так далее.

Длина данной строки не менее 3.

Input	Output
DSGFDFBG	DFB

Г. *Наименьший триплет - 2*

Дана строка, содержащая не менее трёх символов.

Вывести наименьший в лексикографическом порядке триплет данной строки. В этой задаче рассматриваются только триплеты, начинающиеся с индексов, кратных трём.

Длина данной строки не обязательно кратна 3.

Input	Output
DSGFDFBGZA	BGZ

Комментарий к примеру: в данной строке три триплета: DSG, FDF, BGZ. Наименьший из них — триплет BGZ.

Н. *Буквы по одному разу*

Дана ACGT-строка.

Вывести буквы этой строки по одному разу в том порядке, в котором они в этой строке появляются в первый раз.

Input	Output
CCCGGGCCAAACCAATTGGAAC	CGAT
AACCGG	ACG

И. *Комплементарное дополнение*

Дана ACGT-строка.

Вывести её комплементарное дополнение: в развёрнутой задом наперёд строке надо выполнить следующие замены символов: $C \leftrightarrow G$ и $A \leftrightarrow T$.

Input	Output
ACAGCAGT	ACTGCTGT

Ж. *Расстояние Хэмминга - 1*

Расстоянием Хэмминга двух строк одинаковой длины называется количество разных символов, имеющих одинаковые индексы в обеих строках. Например расстояние Хэмминга между строками $s_1 = \text{'ACCGAGT'}$ и $s_2 = \text{'ACAGAGG'}$ равно 2, так как $s_1[2] \neq s_2[2]$ и $s_1[6] \neq s_2[6]$, а остальные символы попарно равны.

Расстояние Хэмминга — один из простейших примеров понятия расстояния на множестве строк.

Даны две непустые ACGT-строки одинаковой длины. Вычислить расстояние Хэмминга между ними.

Input	Output
ACCGAAT ATCGAGT	2
ACGT TGCA	4

К. *Расстояние Хэмминга - 2*

Дана ACGT-строка длины N . Рассмотрим все её циклические сдвиги вправо (на $1, 2, \dots, N - 1$ символов). Вывести $N - 1$ число — расстояния Хэмминга между исходной строкой и каждым её циклическим сдвигом.

Input	Output
ACCACT	5 5 2 5 5

Комментарий к примеру: у данной строки следующие циклические сдвиги:

ТАССАС
СТАССА
АСТАСС
САСТАС
ССАСТА

Соответствующие расстояния Хэмминга равны 5, 5, 2, 5, 5.

L* *Расстояние до ближайшего*

Дана ACGT-строка.

Для каждого символа строки вывести число, равное расстоянию до ближайшего в строке символа, равного данному. Если второго такого символа нет, вывести -1 .

Input	Output
CGCAATACGCAGTCCSSA	2 7 2 1 1 7 2 2 3 2 4 3 7 1 1 1 1 7

M. *Составить строку*

Даны две ACGT-строки S_1 и S_2 . Сколько экземпляров строки S_2 надо взять, чтобы из всех этих символов (возможно, используя только часть из них) можно было составить строку S_1 ?

Вывести одно натуральное число — ответ на вопрос задачи, или число -1 , если сделать это невозможно.

Input	Output
AAAAAAAAGGG ACCATG	4
ACGACGT ACG	-1

N. *Подпоследовательность*

Даны две ACGT-строки S_1 и S_2 . Проверить, что строка S_2 встречается, как подпоследовательность в строке S_1 . То есть символы из строки S_2 встречаются в S_1 в том же порядке, в котором они следуют в строке S_2 , но, возможно, не стоят рядом.

Если это так, вывести YES, иначе вывести NO.

В решении нельзя изменять данные строки и создавать новые.

Input	Output
AACCGGTT ACGT	YES
ACAGCACGTAC TG	NO

O. *Сколько раз одна строка входит 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.

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

Input	Output
AAABAABVBAVBAABVBAVBA AAB	3
AAAAAA AAA	4

P. *Магическая последовательность*

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

По данному натуральному числу K выпишите последовательность с номером K .

Input	Output
4	1211

Q. *Красные и синие - 1*

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
BBBRBRBR	BBBRRR 2

R. *Красные и синие - 2*

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
BVBRBRBRBRBRBBVBRBR	BBBBBBBBBRRR 5

S. *Красные и синие - 3*

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
BVBRBRBRBRBRBRBRBR	BBBBBRBRBRBR 4

T. *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

U. *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

V. Упаковка строки

Будем рассматривать только строчки, состоящие из заглавных латинских букв. Например, рассмотрим строку AAAABCCCCCDDDD. 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 4AB5C4D. 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

W. 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

X. 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