



# Задание №12, Редактор строк

## Теория



**s = "abc"** - объявление строки

**s = "ab" + "cd"** - сложение (конкатенация) строк

**s = "ab" \* 3**  $\Rightarrow$  **s = "ababab"** - умножение строк

**s.replace(str\_old, str\_new, count)** - возвращает изменённую строку **s**, в которой подстроки **str\_old** были заменены на строки **str\_new** указанное в **count** количество раз, считая слева. Важно, эта функция не изменяет исходную строку **s**, поэтому для применения изменений обязательно переприсваиваем переменную: **s = s.replace(...)**

## Соотнесение команд из условие с Python:

ПОКА	WHILE
ЕСЛИ ТО	IF
ИНАЧЕ	ELSE

ИНАЧЕ ЕСЛИ	ELIF
ЗАМЕНИТЬ	REPLACE

## План решения

**Легкий тип. Фиксированная строка на вход**

*Демовариант 2025*

Исполнитель Редактор получает на вход строку цифр и преобразовывает её. Редактор может выполнять две команды, в обеих командах  $v$  и  $w$  обозначают цепочки цифр.

А) **заменить** ( $v, w$ ).

Эта команда заменяет в строке первое слева вхождение цепочки  $v$  на цепочку  $w$ . Например, выполнение команды

**заменить** (111, 27)

преобразует строку 05111150 в строку 0527150.

Если в строке нет вхождений цепочки  $v$ , то выполнение команды **заменить** ( $v, w$ ) не меняет эту строку.

Б) **нашлось** ( $v$ ).

Эта команда проверяет, встречается ли цепочка  $v$  в строке исполнителя Редактор. Если она встречается, то команда возвращает логическое значение «истина», в противном случае возвращает значение «ложь». Строка исполнителя при этом не изменяется.

Цикл

ПОКА *условие*

*последовательность команд*

КОНЕЦ ПОКА

выполняется, пока условие истинно.

В конструкции

ЕСЛИ *условие*

ТО *команда1*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно).

В конструкции

ЕСЛИ *условие*

ТО *команда1*

ИНАЧЕ *команда2*

КОНЕЦ ЕСЛИ

выполняется *команда1* (если условие истинно) или *команда2* (если условие ложно).

Какая строка получится в результате применения приведённой ниже программы к строке, состоящей из 81 идущей подряд цифры 1? В ответе запишите полученную строку.

НАЧАЛО

ПОКА **нашлось** (11111) ИЛИ **нашлось** (888)

ЕСЛИ **нашлось** (11111)

ТО **заменить** (11111, 88)

ИНАЧЕ **заменить** (888,8)

КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

1. Записываем переменную, заданную в условии.

```
a = 81 * '1'
```

2. Переписываем алгоритм из условия.

```
while '11111' in a or '888' in a:  
    if '11111' in a:  
        a = a.replace('11111', '88', 1)  
    else:  
        a = a.replace('888', '8')
```

3. Выводим результат.

```
print(a)
```

Получаем ответ: **881**. Весь код:

```
a = 81 * '1'  
while '11111' in a or '888' in a:  
    if '11111' in a:  
        a = a.replace('11111', '88', 1)  
    else:  
        a = a.replace('888', '8')  
print(a)
```

**Тип посложнее. Перебор строк с разным количеством цифр**  
**6A308D**

Дана программа для Редактора:

НАЧАЛО

ПОКА нашлось (19) ИЛИ нашлось (49) ИЛИ нашлось (999)

    ЕСЛИ нашлось (19)

        ТО заменить (19, 9)

    КОНЕЦ ЕСЛИ

    ЕСЛИ нашлось (49)

        ТО заменить (49, 91)

    КОНЕЦ ЕСЛИ

    ЕСЛИ нашлось (999)

        ТО заменить (999, 4)

    КОНЕЦ ЕСЛИ

КОНЕЦ ПОКА

КОНЕЦ

На вход приведённой выше программе поступает строка, начинающаяся с цифры «1», а затем содержащая  $n$  цифр «9» ( $3 < n < 10\,000$ ).

Определите **наибольшее** возможное значение суммы числовых значений цифр в строке, которая может быть результатом выполнения программы.

1. Объявляем цикл перебора переменной, которая будет отвечать за количество тех или иных цифр в строке. Допустимый диапазон обязательно берём из условия.

```
for n in range(3,10000):  
    x = '1' + n * '9'
```

2. Переписываем алгоритм из условия, аналогично первому типу.

```
while '19' in x or '49' in x or '999' in x:  
    if '19' in x:  
        x = x.replace('19','9',1)  
    if '49' in x:  
        x = x.replace('49','91',1)  
    if '999' in x:  
        x = x.replace('999', '4', 1)
```

3. Пишем условие к изменённому  $x$ . Чаще всего встречаются:

- a. Подсчитайте количество таких **n**, при которых **x** после работы алгоритма выполняет следующее условие:

```
cnt = 0
for n in range(...):
    ... # алгоритм
    if (условие):
        cnt+=1
print(cnt)
```

- b. Определите наименьшее/наибольшее **n**, при котором **x** после работы алгоритма выполняет следующее условие:

```
for n in range(...):
    ... # алгоритм
    if (условие):
        print(n)
```

- c. Выведите наибольшие/наименьше возможные сумму цифр/числовое значение/длину/... **x**:

```
max_x = -1
for n in range(...):
    ... # алгоритм
    if x > max_x:
        max_x = x
print(max_x)
```

4. В конкретно данном примере нужно найти максимальную сумму цифр, поэтому вводим для этого переменную **max\_sum**, которая работает по принципу установки рекорда. Функцией **map** применяем **int** к каждому символу полученной строки и считаем сумму функцией **sum**. Если эта сумма оказывается больше, чем предыдущий максимум, то обновляем переменную. Выводим переменную после окончания цикла и получаем ответ **23**. Весь код:

```
max_sum = -1
for n in range(3,10000):
    x = '1' + n * '9'
    while '19' in x or '49' in x or '999' in x:
        if '19' in x:
            x = x.replace('19','9',1)
        if '49' in x:
            x = x.replace('49','91',1)
        if '999' in x:
            x = x.replace('999', '4', 1)
    sum_x = sum(map(int,x))
    if sum_x > max_sum:
        max_sum = sum_x
print(max_sum)
```