

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №2.3**

Работа со строками в языке Python

Выполнил студент группы

ИВТ-б-о-21-1

Мальцев Н.А. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2022

## Работа со строками в языке Python.

**Цель работы:** приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.10.

### Порядок выполнения работы:

1) Проработка примеров.

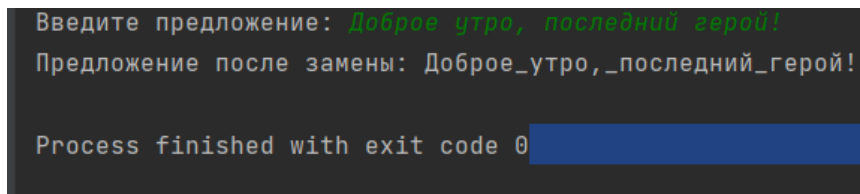
Пример 1.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    s = input("Введите предложение: ")
    r = s.replace(' ', '_')
    print(f'Предложение после замены: {r}')
```

Результат работы программы:



```
Введите предложение: Доброе утро, последний герой!
Предложение после замены: Доброе_утро,_последний_герой!

Process finished with exit code 0
```

Рисунок 1. Результат выполнения программы из первого примера

Пример 2.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

if __name__ == '__main__':
    word = input("Введите слово: ")

    idx = len(word) // 2
    if len(word) % 2 == 1:
        # Длина слова нечетная.
        r = word[:idx] + word[idx+1:]
    else:
        # Длина слова четная.
        r = word[:idx-1] + word[idx+1:]
    print(r)
```

Результат работы программы:

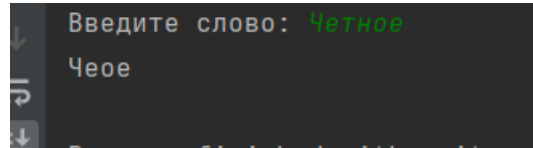


Рисунок 2. Результат работы программы из примера 2

Пример 3.

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    n = int(input("Введите длину: "))

    # Проверить требуемую длину.
    if len(s) >= n:
        print(
            "Заданная длина должна быть больше длины предложения",
            file=sys.stderr
        )
        exit(1)

    # Разделить предложение на слова.
    words = s.split(' ')
    # Проверить количество слов в предложении.
    if len(words) < 2:
        print(
            "Предложение должно содержать несколько слов",
            file=sys.stderr
        )
        exit(1)

    # Количество пробелов для добавления.
    delta = n
    for word in words:
        delta -= len(word)

    # Количество пробелов на каждое слово.
    w, r = delta // (len(words) - 1), delta % (len(words) - 1)

    # Сформировать список для хранения слов и пробелов.
    lst = []

    # Пронумеровать все слова в списке и перебрать их.
    for i, word in enumerate(words):
        lst.append(word)

    # Если слово не является последним, добавить пробелы.
    if i < len(words) - 1:
        # Определить количество пробелов.
        width = w
        if r > 0:
            width += 1
```

```

    r -= 1

    # Добавить заданное количество пробелов в список.
    if width > 0:
        lst.append(' ' * width)

    # Вывести новое предложение, объединив все элементы списка lst.
    print(''.join(lst))

```

Результат работы программы:

```

"C:\Users\Николай Мальцев\AppData\Local\Progra
Введите предложение: Пример работы программы
Введите длину: 100
Примерработыпрограммы
Process finished with exit code 0

```

Рисунок 3. Результат работы программы из примера 3

## 2) Выполнение индивидуального задания. (Вариант 9)

Задание 1. Дано предложение. Вывести столбиком его третий, шестой и т.д. символы.

Код программы:

```

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

text = input("Введите предложение: ")
for i in range(2, len(text), 3):
    print(text[i])

```

Результат выполнения программы:

```

Введите предложение: Перенос требуют наши сердца
р
е
т
б
т
а
р
а

```

Рисунок 4. Результат работы программы для решения индивидуального задания 1

Задание 2. Дано предложение. Определить, есть ли в нем словосочетания «чу» или «щу». В случае положительного ответа найти также порядковый номер первой буквы первого из них.

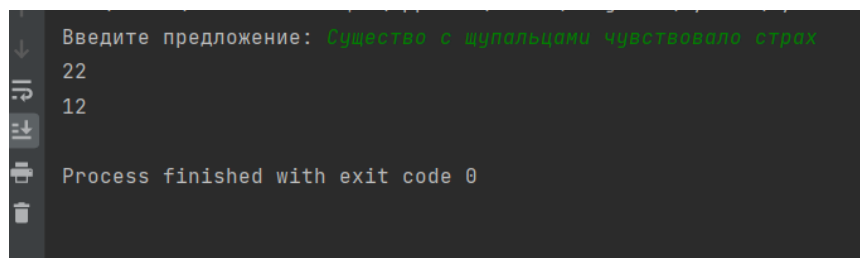
Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

s = input("Введите предложение: ")

if 'чу' in s:
    print(s.find('чу'))
if 'щү' in s:
    print(s.find('щү')+1)
```

Результат выполнения программы:



```
Введите предложение: Существо с щупальцами чувствовало страх
22
12
Process finished with exit code 0
```

Рисунок 5. Результат выполнения программы к индивидуальному заданию 2

Задание 3. Дано слово, оканчивающееся символом «.». Составить программу, которая вставляет некоторую заданную букву после буквы с заданным номером.

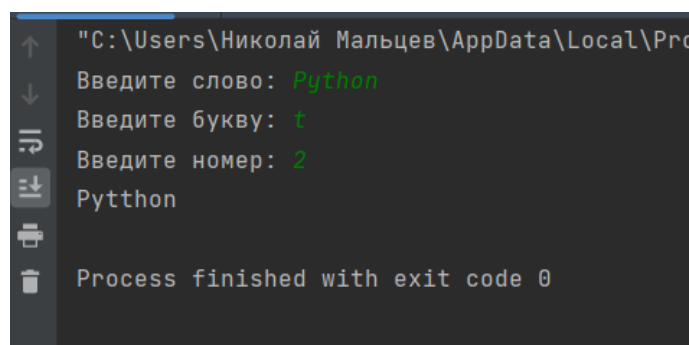
Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

word = input("Введите слово: ")
char = input("Введите букву: ")
num = int(input("Введите номер: "))

s = (word[0:num]+char+word[num:])
print(s)
```

Результат выполнения программы:



```
"C:\Users\Николай Мальцев\AppData\Local\Pro
Введите слово: Python
Введите букву: t
Введите номер: 2
Pytthon
Process finished with exit code 0
```

Рисунок 6. Результат выполнения программы к индивидуальному заданию 3

### 3) Выполнение задания повышенной сложности.

Вариант 9. Даны два слова. Определить, можно ли из букв первого из них получить второе.

Рассмотреть два варианта:

1. Повторяющиеся буквы второго слова могут в первом слове не повторяться:

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

word_1 = input("Введите первое слово: ")
word_2 = input("Введите второе слово: ")

word_1 = list(set(word_1.lower()))
word_2 = list(set(word_2.lower()))

list_word1 = list(word_1)
list_word2 = list(word_2)

list_word1.sort()
list_word2.sort()

if list_word1 == list_word2:
    print("Можно")
else:
    print("Нельзя")
```

Результат выполнения программы:

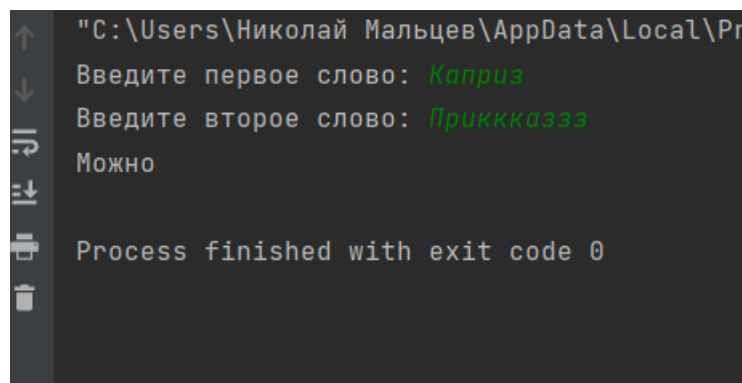


Рисунок 7. Результат выполнения программы к усложненному заданию (версия 1)

2. Каждая буква второго слова должна входить в первое слово столько же раз, сколько и во второе:

Код программы:

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

word_1 = input("Введите первое слово: ")
```

```

word_2 = input("Введите второе слово: ")

word_1 = word_1.lower()
word_2 = word_2.lower()

list_word1 = list(word_1)
list_word2 = list(word_2)

list_word1.sort()
list_word2.sort()

if list_word1 == list_word2:
    print("Можно")
else:
    print("Нельзя")

```

Результат выполнения программы:

```

"C:\Users\Николай Мальцев\AppData\Local\Python\Python310\python.exe"
Введите первое слово: Каприз
Введите второе слово: Приказ
Можно
Process finished with exit code 0

```

Рисунок 8. Результат выполнения программы к усложненному заданию (версия 2)

```

"C:\Users\Николай Мальцев\AppData\Local\Python\Python310\python.exe"
Введите первое слово: Каприз
Введите второе слово: Капитан
Нельзя
Process finished with exit code 0

```

Рисунок 9. Результат выполнения программы к усложненному заданию (вариант работы)

**Вывод:** в ходе работы был приобретен навыки работы со строками при программировании на языке Python версии 3.10.