

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Кафедра инфокоммуникаций

**«Исследование возможностей Git для работы с
локальными репозиториями»**

**Отчет по лабораторной работе № 2.3
по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-21-1

Степанов Д.А. .«14» сентября 2022г.

Подпись студента_____

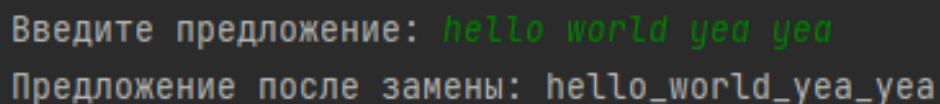
Работа защищена « »_____20__г.

Проверила Воронкин Р.А. _____
(подпись)

МЕТОДИКА И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

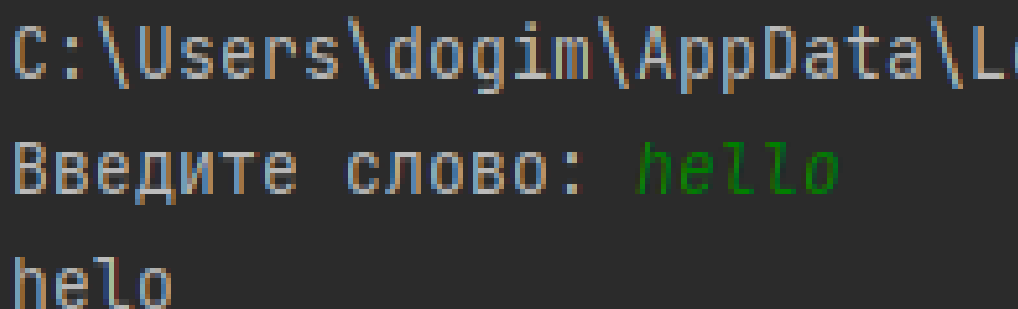
№1

Проработайте примеры лабораторной работы. Создайте для каждого примера отдельный модуль языка Python. Зафиксируйте изменения в репозитории.



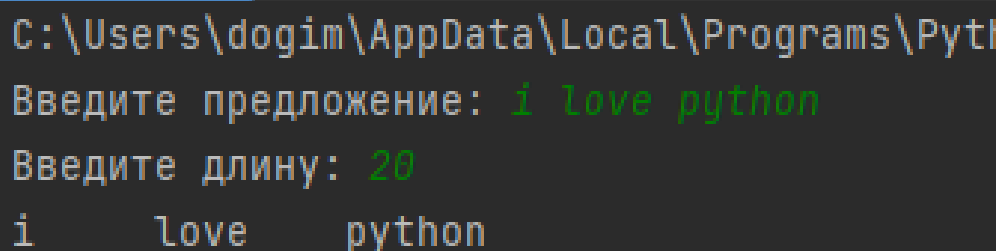
```
Введите предложение: hello world yea yea
Предложение после замены: hello_world_yea_yea
```

Результат выполнения первого примера



```
С:\Users\dogim\AppData\Local\Programs\Python\Python38\python.exe
Введите слово: hello
helo
```

Результат выполнения второго примера



```
С:\Users\dogim\AppData\Local\Programs\Python\Python38\python.exe
Введите предложение: i love python
Введите длину: 20
i love python
```

Результат выполнения третьего примера

№2

Выполните индивидуальные задания, согласно своему варианту. Для заданий повышенной сложности номер варианта должен быть получен у преподавателя.

Индивидуальные задания

Задание 1

Дано предложение. Вывести все буквы м и н в нем.

```
m C:\ 1 ▶ #!/usr/bin/env python3
nple_ 2 # -*- coding: utf-8 -*-
nple_ 3
nple_ 4 ▶ if __name__ == '__main__':
vidual 5     s = input("Введите предложение: ")
Libra 6
es and 7     if s.count('м') or s.count('н'):
8         print(s.count('м') * 'м' + s.count('н') * 'н')
9     else:
10         print("Букв м и н нет в этом предложении!")
if __name__ == '__main__': > else
```

individual_task_1 ×

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe
Введите предложение: я пишу это сообщение
мнн

Задание 2

Дана последовательность слов. Проверить, правильно ли в ней записаны буквосочетания ча и ща. Исправить ошибки.

```
m C:\ 1 ▶ #!/usr/bin/env python3
nple_ 2 # -*- coding: utf-8 -*-
nple_ 3
nple_ 4 ▶ if __name__ == '__main__':
vidual 5     s = input("Введите последовательность слов: ")
vidual 6
Libra 7     new_s = s.replace("чя", "ча")
es and 8     new_s = new_s.replace("щя", "ща")
9
10     if new_s == s:
11         print("Вы написали последовательность слов правильно!!!")
12     else:
13         print(f"Исправленная последовательность слов -> {new_s}")
if __name__ == '__main__':
```

individual_task_2 ×

C:\Users\dogim\AppData\Local\Programs\Python\Python310\python.exe C:\0PI\soft
Введите последовательность слов: чя чя чя щя щя щя ща ща
Исправленная последовательность слов -> ча ча ча ща ща ща ща ща

Задание 3

Дано предложение, оканчивающее символом «.». Вставить заданную букву перед последней буквой и.

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

import sys

if __name__ == '__main__':
    s = input("Введите предложение: ")
    letter = input("Введите букву, которая будет стоять перед и: ")

    if len(letter) > 1:
        print(
            "Нужно было ввести одну букву!!",
            file=sys.stderr
        )
        exit(1)

    if s[-1] != '.':
        print(
            "В конце предложения нужно было поставить точку!!!",
            file=sys.stderr
        )
        exit(1)

    temp = s.rfind('и')
    if temp == -1:
        print(
            "Буква 'и' не была найдена",
            file=sys.stderr
        )
        exit(1)

    new_s = s[:temp] + letter + s[temp:]

    print(new_s)
```

Введите предложение: я люблю пИтон.

Введите букву, которая будет стоять перед и: и
я люблю пИтон.

Задача повышенной сложности

Даны три слова. Напечатать их общие буквы. Повторяющиеся буквы каждого слова не рассматривать.

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

import sys

if __name__ == '__main__':
    words = input("Введите три слова: ").lower().split()

    # Выдает ошибку если не 3 слова
    if len(words) > 3 or len(words) < 3:
        print(
            "Введено не три слова",
            file=sys.stderr
        )
        exit(1)

    # Находим повторяющиеся буквы
    letters = set()
    for i in words[0]:
        if (i not in letters) and (i in words[1]) and (i in words[2]):
            letters.add(i)

    # Выводим повторяющиеся буквы
    if len(letters) == 0:
        print("Общих букв нет")
    else:
        print(f"Буквы которые повторяются -> {' '.join(letters)}")
```

```
Введите три слова: про ролтон ролики
Буквы которые повторяются -> о, р
```

Вопросы для защиты работы

1. Что такое строки в языке Python?

Строки в Python - упорядоченные последовательности символов, используемые для хранения и представления текстовой информации, поэтому с помощью строк можно работать со всем, что может быть представлено в текстовой форме.

2. Какие существуют способы задания строковых литералов в языке Python?

Строки в апострофах и в кавычках, экранированные последовательности, "сырые" строки, строки в тройных апострофах или кавычках

3. Какие операции и функции существуют для строк?

Сложение, дублирование, длина строки, длина строки, извлечение среза и т. д.

4. Как осуществляется индексирование строк?

Доступ к символам в строках основан на операции индексирования — после строки или имени переменной, ссылающейся на строку, в квадратных скобках указываются номера позиций необходимых символов.

5. Как осуществляется работа со срезами для строк?

Есть три формы срезов. Самая простая форма среза: взятие одного символа строки, а именно, `S[i]` — это срез, состоящий из одного символа, который имеет номер `i`, при этом считая, что нумерация начинается с числа 0. То есть если `S = 'Hello'`, то `S[0]=='H'`, `S[1]=='e'`, `S[2]=='l'`, `S[3]=='l'`, `S[4]=='o'`.

Если указать отрицательное значение индекса, то номер будет отсчитываться с конца, начиная с номера -1.

Срез с двумя параметрами: `S[a:b]` возвращает подстроку из `b-a` символов, начиная с символа с индексом `a`, то есть до символа с индексом `b`, не включая его.

6. Почему строки Python относятся к неизменяемому типу данных?

Строки — один из типов данных, которые Python считает неизменяемыми, что означает невозможность их изменять. Python дает возможность изменять (заменять и перезаписывать) строки.

7. Как проверить то, что каждое слово в строке начинается с заглавной буквы?
`string.istitle()`

8. Как проверить строку на вхождение в неё другой строки?
`string.find()`

9. Как найти индекс первого вхождения подстроки в строку?
`s.partition(<sep>)`

10. Как подсчитать количество символов в строке?
`len(s)`

11. Как подсчитать то, сколько раз определённый символ встречается в строке?
`s.count(<sub>)`

12. Что такое f-строки и как ими пользоваться?

Эти строки улучшают читаемость кода, а также работают быстрее чем другие способы форматирования. F-строки задаются с помощью литерала «f» перед кавычками. Пример: `print(f"Меня зовут {name} Мне {age} лет.")`

13. Как найти подстроку в заданной части строки?
`s.find(значение, начало, конец)`

14. Как вставить содержимое переменной в строку, воспользовавшись методом `format()`?
`print('{}'.format(s))`

15. Как узнать о том, что в строке содержатся только цифры?
`s.isdigit()`

16. Как разделить строку по заданному символу?

`str.split()`

17. Как проверить строку на то, что она составлена только из строчных букв?

`s.isalpha()`

18. Как проверить то, что строка начинается со строчной буквы?

`s.istitle()`

19. Можно ли в Python прибавить целое число к строке?

Нет

20. Как «перевернуть» строку?

`s.reverse()`

21. Как объединить список строк в одну строку, элементы которой разделены дефисами?

`str.split('-')`

22. Как привести всю строку к верхнему или нижнему регистру?

`s.upper()`
`s.lower`

23. Как преобразовать первый символ строки к верхнему регистру?

`s.capitalize()`

24. Как проверить строку на то, что она составлена только из прописных букв?

`s.isupper()`

25. В какой ситуации вы воспользовались бы методом `splitlines()`?

`s.splitlines()` делит `s` на строки и возвращает их в списке. Любой из следующих символов или последовательностей символов считается границей строки.

26. Как в заданной строке заменить на что-либо все вхождения некоей подстроки?

`s.replace(old, new)`

27. Как проверить то, что строка начинается с заданной последовательности символов, или заканчивается заданной последовательностью символов?

`str.startswith()` и `str.endswith()`

28. Как узнать о том, что строка включает в себя только пробелы?

`s.isspace()`

29. Что случится, если умножить некую

строку на 3? `AQ*3 = AQAQAQ`

30. Как привести к верхнему регистру первый символ каждого слова в строке?

`s.title()`

31. Как пользоваться методом `partition()`?

Метод `partition()` разбивает строку при первом появлении строки аргумента и возвращает кортеж, содержащий часть перед разделителем, строку аргумента и часть после разделителя.

32. В каких ситуациях пользуются методом `rfind()`?

`s.rfind(<sub>)` возвращает индекс последнего вхождения подстроки `<sub>` в `s`, который соответствует началу `<sub>`.