Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6**

**дисциплины «Программирование на Python»**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Лейс Алексей Вячеславович  2 курс, группа ИВТ-б-о-22-1,  09.03.01 «Информатика и вычислительная техника», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики: кандидат тех. наук доцент кафедры инфокоммуникаций: Воронкин Р.А  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2022 г.

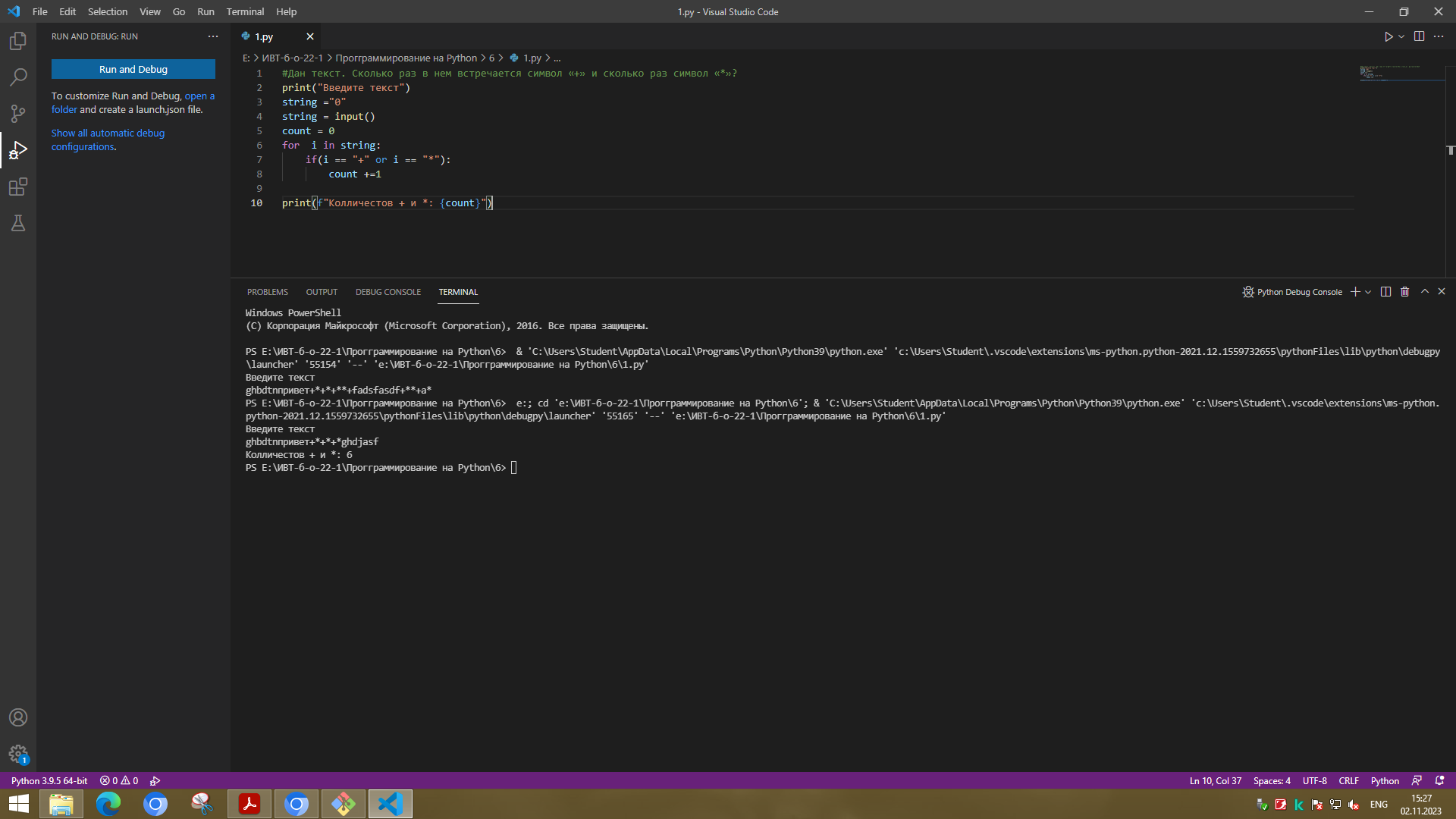
**Тема:** Работа со строками в языке Python

**Цель работы:** приобретение навыков по работе со строками при написании программ с помощью языка программирования Python версии 3.x.

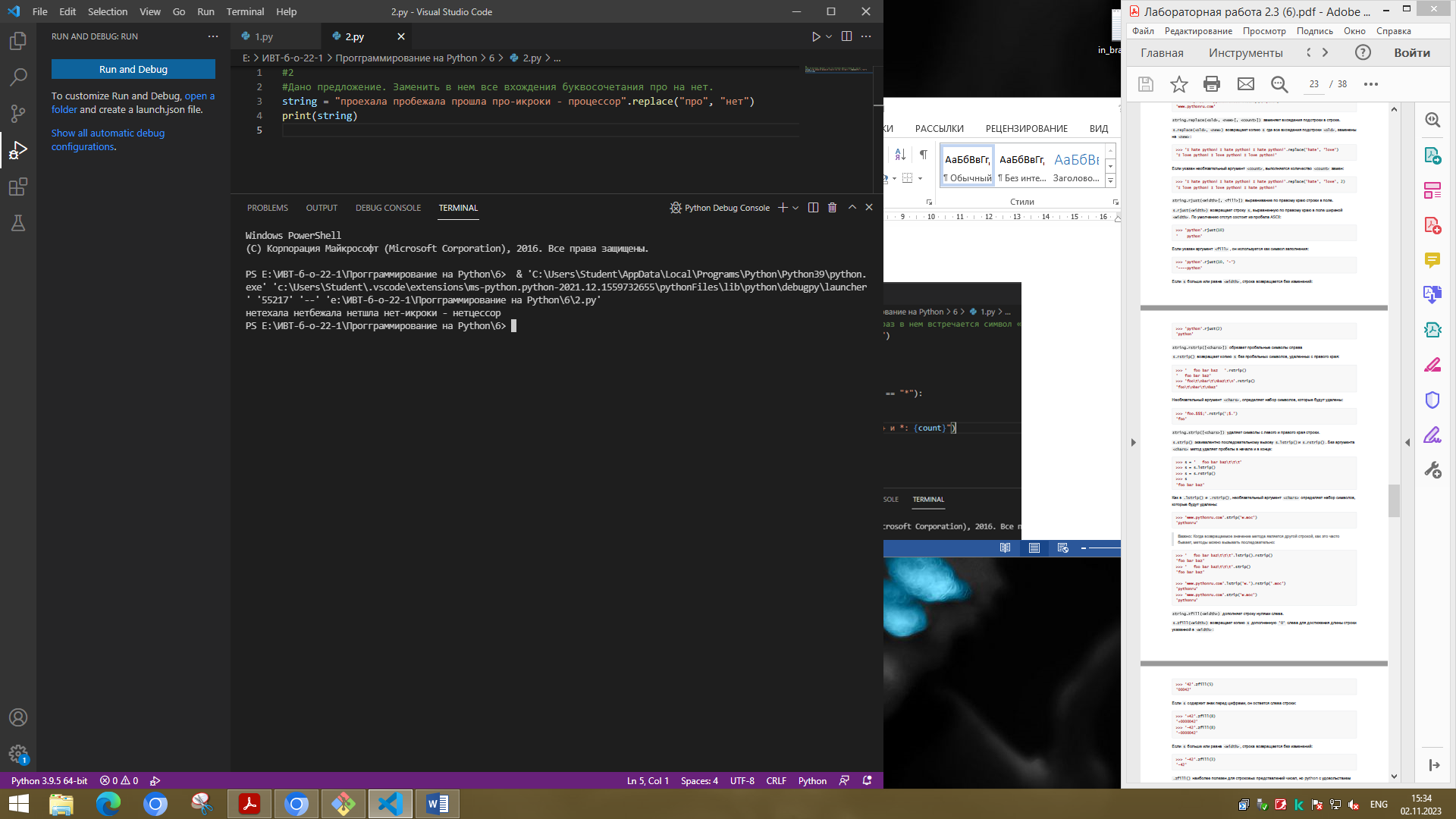
**Порядок выполнения работы:**

Индивидуальное задание:

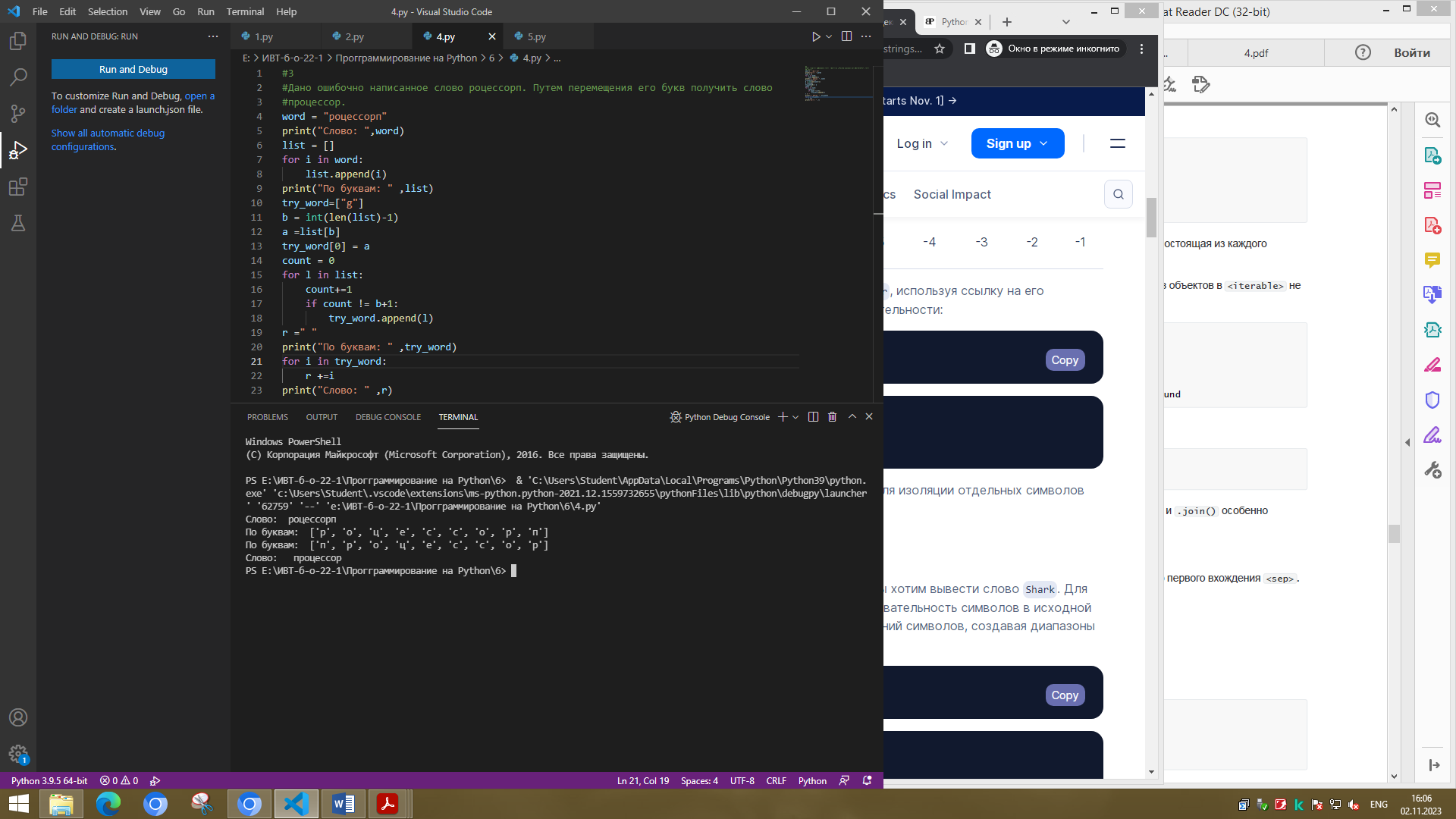
Дано предложение. Определить, сколько в нем одинаковых соседних букв.



Дано предложение. Заменить в нем все вхождения буквосочетания про на нет.

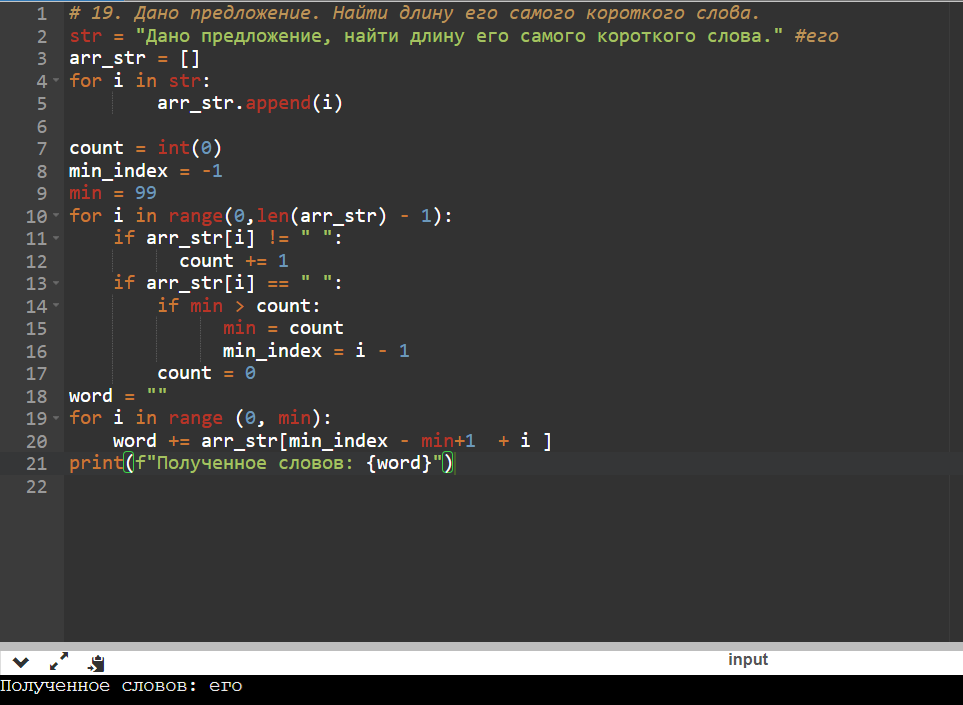


Дано ошибочно написанное слово роцессорп. Путем перемещения его букв получить слово процессор.



Повышенной сложности:

Дано предложение. Найти длину его самого короткого слова.



Ответы на вопросы:

1. Что такое строки в языке Python?
   * Строки в Python представляют собой последовательность символов и используются для представления текстовой информации.
2. Какие существуют способы задания строковых литералов в языке Python?
   * Строковые литералы могут быть заданы с использованием одинарных (' '), двойных (" "), или тройных (''' ''' или """ """) кавычек.
3. Какие операции и функции существуют для строк?
   * Существует множество операций и методов для строк, таких как конкатенация (+), умножение (\*), функции len(), методы upper(), lower(), и другие.
4. Как осуществляется индексирование строк?
   * Индексирование строк начинается с 0, и символ в строке может быть получен по его индексу, например, my\_string[0] вернет первый символ строки.
5. Как осуществляется работа со срезами для строк?
   * Срезы в Python создаются с использованием оператора :. Например, my\_string[1:4] возвращает подстроку с 1-го по 3-й символ строки.
6. Почему строки Python относятся к неизменяемому типу данных?
   * Строки являются неизменяемыми, потому что после создания строки нельзя изменить ее содержимое. Можно создать новую строку с необходимыми изменениями.
7. Как проверить то, что каждое слово в строке начинается с заглавной буквы?
   * Можно воспользоваться методом istitle(), который возвращает True, если каждое слово начинается с заглавной буквы.
8. Как проверить строку на вхождение в неё другой строки?
   * Можно воспользоваться оператором in. Например, substring in my\_string вернет True, если подстрока присутствует в строке.
9. Как найти индекс первого вхождения подстроки в строку?
   * Метод find() возвращает индекс первого вхождения подстроки, или -1, если подстрока не найдена.
10. Как подсчитать количество символов в строке?
    * Функция len(my\_string) возвращает количество символов в строке.
11. Как подсчитать то, сколько раз определённый символ встречается в строке?
    * Метод count() позволяет подсчитать количество вхождений определенного символа в строке.
12. Что такое f-строки и как ими пользоваться?
    * F-строки - это способ форматирования строк, позволяющий вставлять значения переменных в строку прямо в месте её объявления. Используется префикс f перед строкой.
13. Как найти подстроку в заданной части строки?
    * Можно воспользоваться методом find() с указанием начального и конечного индексов.
14. Как вставить содержимое переменной в строку, воспользовавшись методом format()?
    * С использованием метода format(), переменные могут быть вставлены в строку. Например, "Привет, {}!".format(name).
15. Как узнать о том, что в строке содержатся только цифры?
    * Метод isdigit() возвращает True, если все символы строки являются цифрами.
16. Как разделить строку по заданному символу?
    * Метод split() разделяет строку на подстроки, используя заданный разделитель.
17. Как проверить строку на то, что она составлена только из строчных букв?
    * Метод islower() возвращает True, если все буквы в строке строчные.
18. Как проверить то, что строка начинается со строчной буквы?
    * Метод islower() также может быть использован для проверки, что первая буква строки строчная.
19. Можно ли в Python прибавить целое число к строке?
    * Нет, строки и числа не могут быть сложены напрямую. Необходимо преобразовать число в строку или наоборот.
20. Как «перевернуть» строку?
    * С использованием среза [::-1], например, reversed\_string = my\_string[::-1].
21. Как объединить список строк в одну строку, элементы которой разделены дефисами?
    * Метод join() может быть использован: "-".join(my\_list).
22. Как привести всю строку к верхнему или нижнему регистру?
    * Методы upper() и lower() соответственно изменяют строку на верхний и нижний регистр.
23. Как преобразовать первый и последний символы строки к верхнему регистру?
    * Можно воспользоваться методами capitalize() для первого символа и title() для каждого слова.
24. Используя методы `capitalize()` для первого символа и `upper()` для последнего:

```python

string = string[0].capitalize() + string[1:-1] + string[-1].upper()

```

25. С помощью метода `islower()`:

```python

is\_lower = my\_string.islower()

```

26. Метод `splitlines()` используется для разделения строки на список строк по символу новой строки.

27. С использованием метода `replace()`:

```python

new\_string = original\_string.replace(substring\_to\_replace, replacement\_string)

```

28. Для проверки начала или конца строки используются методы `startswith()` и `endswith()` соответственно.

29. Чтобы проверить, что строка состоит только из пробелов, используйте метод `isspace()`:

```python

contains\_only\_spaces = my\_string.isspace()

```

30. Строка будет повторена три раза.

31. Используя метод `title()`:

```python

new\_string = original\_string.title()

```

32. Метод `partition()` разделяет строку на три части по первому вхождению указанной подстроки и возвращает кортеж.

33. Метод `rfind()` используется для поиска последнего вхождения подстроки в строке и возвращает индекс этого вхождения.

Вывод: на основе выполненной работы приобрели навыки по работе со строками при написании программ с помощью языка программирования Python версии 3.x.