**РЕСПУБЛИКА БЕЛАРУСЬ**

**КОМИТЕТ ПО ОБРАЗОВАНИЮ МИНГОРИСПОЛКОМА**

**учреждение образования**

**МИНСКИЙ ГОСУДАРСТВЕННЫЙ КОЛЛЕДЖ ЦИФРОВЫХ**

**ТЕХНОЛОГИИ**

**Группа 81ТП**

**Отчет**

**по лабораторной работе № 5**

**“Разработка и отладка алгоритмов работы со строками**

**”**

Учебный предмет

«Инструментальное программное обеспечение»

**Исполнитель: Житкевич М.Д.**

**Руководитель: Бровка Д.С.**

Минск, 2024

**Цели:**

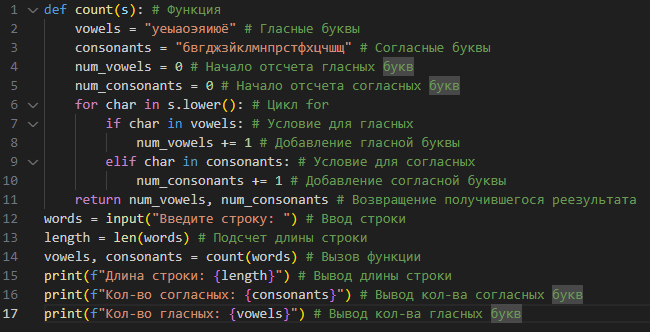
1. Изучение основ программирования на Python.
2. Развитие навыков работы с базовыми синтаксическими конструкциями.
3. Углубление знаний о функциях, модулях и различных типах данных.

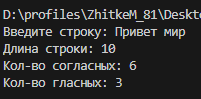
**Задачи:**

1. Ознакомление с синтаксисом языка Python.
2. Освоение создания и использования функций и модулей.
3. Практическое применение условных операторов и циклов.
4. Понимание основ работы с коллекциями данных.
5. Решение практических задач на основе изученных концепций.

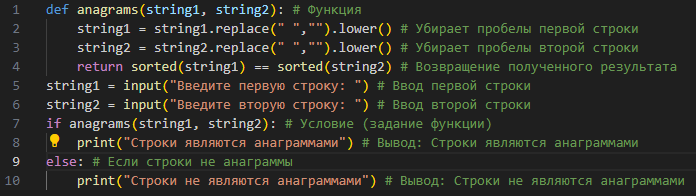
Вариант 1

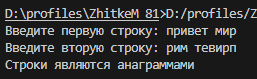




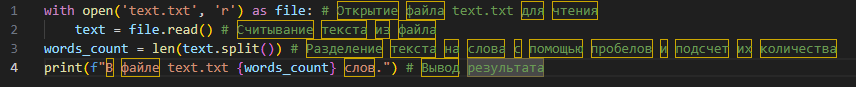




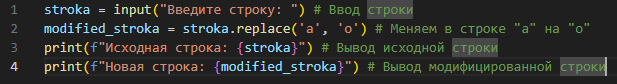


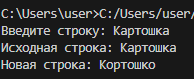




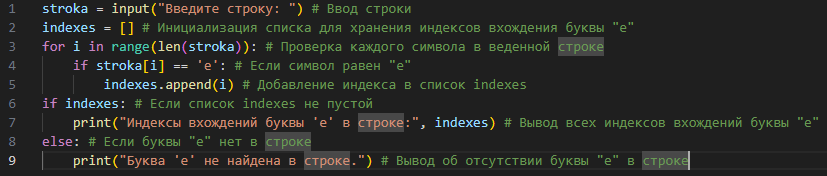


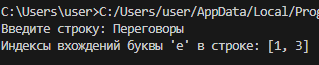




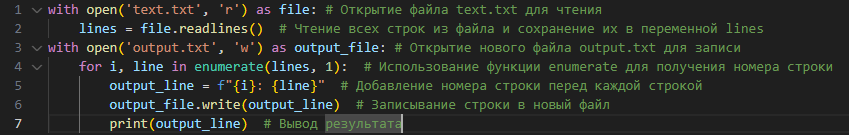












**Ответы на контрольные вопросы:**

1. Строка в Python - это последовательность символов, которая заключается в одинарные или двойные кавычки.

2. Длину строки можно получить с помощью функции len(), например, len(my\_string).

3. Для чтения содержимого файла в Python используется функция open() в режиме чтения, например, with open('file.txt', 'r') as file.

4. Для записи данных в файл в Python также используется функция open(), но в режиме записи, например, with open('file.txt', 'w') as file.

5. Первый символ строки можно извлечь с помощью индексации, например, my\_string[0].

6. Для замены определенных символов в строке используется метод replace(), например, my\_string.replace('a', 'b').

7. Методы strip(), lower() и upper() выполняют соответственно удаление пробелов по краям строки, преобразование строки в нижний регистр и верхний регистр.

8. Для нахождения количества слов в строке можно использовать метод split(), а затем функцию len().

9. Все символы строки можно перебрать с помощью цикла for, например, for char in my\_string:.

10. Для объединения списка строк в одну строку можно использовать метод join(), например, ' '.join(my\_list).

11. Метод split() разбивает строку на подстроки по заданному разделителю и возвращает список подстрок.

12. Для проверки наличия подстроки в строке можно использовать оператор in, например, if 'substring' in my\_string:.

13. Если метод replace() не найдет заменяемый символ в строке, то строка останется неизменной.

14. Для замены одной буквы на другую в строке можно воспользоваться методом replace(), например, my\_string.replace('a', 'b').

15. Для подсчета количества раз, которое слово встречается в тексте, можно использовать метод count(), например, my\_string.count('word').

16. Для разворота строки с использованием срезов можно использовать конструкцию [::-1], например, reversed\_string = my\_string[::-1].

17. Существует два основных способа работы с текстовыми файлами в режиме чтения и записи: использование функции open() в соответствующем режиме или работа с файлами с помощью модуля pathlib.