Задание 1

Пользователь вводит с клавиатуры строку.

- Произведите поворот строки.

- Посчитайте количество букв, цифр в строке.

Полученный результат выведите на экран.

Задание 2

Пользователь вводит с клавиатуры строку и символ или слово для поиска. Посчитайте сколько раз в строке встречается этот символ или слово,

Полученный результат выведите на экран.

Задание 3

Пользователь вводит с клавиатуры строку, слово для поиска, слово для замены. Произведите в строке замену одного слова на другое.

Полученную строку отобразите на экране.

Задание 4

Есть некоторый текст.

Реализуйте следующую функциональность:

■ Изменить текст таким образом, чтобы каждое предложение начиналось с большой буквы;

■ Посчитайте сколько раз цифры встречаются в тексте;

■ Посчитайте сколько раз знаки препинания встречаются в тексте;

■ Посчитайте количество восклицательных знаков в тексте.

Полученный результат выведите на экран.

text = '''

В Python 3.12 появилась поддержка perf profiling. В этой статье

рассмотрим, как это помогает сократить время выполнения Python-

скрипта с 36 секунд до 0,8! мы рассмотрим Linux-инструмент perf, а

также графики Flame Graph (добавить пояснение: способ визуализации

процессорного времени, потраченного на функции), посмотрим на

дизассемблированный код и займемся поиском ошибок.

загляните на соответствующую страницу официальной документации

Python и в список изменений. Для этой статьи из документов по

ссылкам выше важно следующее:

Профилировщик perf для Linux является мощным инструментом,

который позволяет профилировать и получать информацию о

производительности приложения. У perf богатая экосистема

инструментов, которые помогают с анализом данных, которые он

производит.

основная проблема при использовании профилировщика perf с

приложениями Python состоит в том, что perf позволяет получить

информацию только о нативных символах, то есть об именах функций

и процедур, написанных на C. Это значит, что имена и названия файлов

функций Python в вашем коде в выводе perf не появятся.

начиная с Python 3.12, интерпретатор может работать в специальном

режиме, который позволяет функциям Python появляться в выводе

профилировщика perf! при включенном режиме интерпретатор

вставляет небольшой фрагмент кода, скомпилированный на лету, перед

выполнением каждой функции Python и обучает perf взаимосвязи

между этим фрагментом кода и связанной с ним функцией Python с

помощью файлов perf map.

'''