Лабораторная работа №3

**Тема**: Стандартные типы данных, коллекции, функции, модули.

**Цель**: освоить базовый синтаксис языка Python, приобрести навыки работы со стандартными типами данных, коллекциями, функциями, модулями и закрепить их на примере разработки интерактивных приложений.

**Cсылки**:

1. <https://docs.python.org/>
2. <https://pythonworld.ru/>
3. <https://smartiqa.ru/courses/python/lesson-1>
4. <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>
5. <https://pythontutor.ru/>
6. <https://pythonchik.ru/osnovy>
7. <https://younglinux.info/python/course>
8. <https://www.w3schools.com/python/default.asp>

Для защиты ЛР необходимо оформить Отчет со скринами кода и результатов его выполнения

**Требования к выполнению**

1. Программа должна быть снабжена комментариями на английском языке, в которых необходимо указать краткое предназначение программы, номер лабораторной работы и название, версию программы, Ф.И.О. разработчика и дату разработки.
2. Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом
3. Выполнить документирование кода для получения справки по каждой функции
4. Каждое задание оформить в виде отдельной бизнес-функции.
5. При разработке программ рекомендуется придерживаться принципа: за решение одной конкретной задачи должна отвечать одна функция.
6. Все функции необходимо сгруппировать в модулях, согласно их логике их работы.
7. Разработанные основные функции, размещенные в отдельных модулях, нужно подключить в другом модуле, где будет происходить тестирование данных функций.
8. Размерность списка задается пользователем.
9. Предусмотреть способы инициализации последовательности: с помощью **функции** **генератора** и пользовательского ввода. Оформить способы инициализации в виде отдельных функций, которые на вход принимают последовательность для инициализации, и сгруппировать эти функции в отдельный модуль от основной функции программы.
10. Продемонстрировать использование **декоратора** в любом из заданий
11. В программах предусмотреть возможность повторного выполнения без выхода из программы и защиту от ввода некорректных пользовательских данных. Для этих целей рекомендуется разработать отдельные функции.
12. Обеспечить обработку конкретных классов исключений

**Индивидуальные задания по вариантам**

**Задание 1.** В соответствии с заданием своего варианта составить программу для вычисления значения функции c помощью разложения функции в степенной ряд. Задать точность вычислений eps.

Предусмотреть максимальное количество итераций, равное 500.

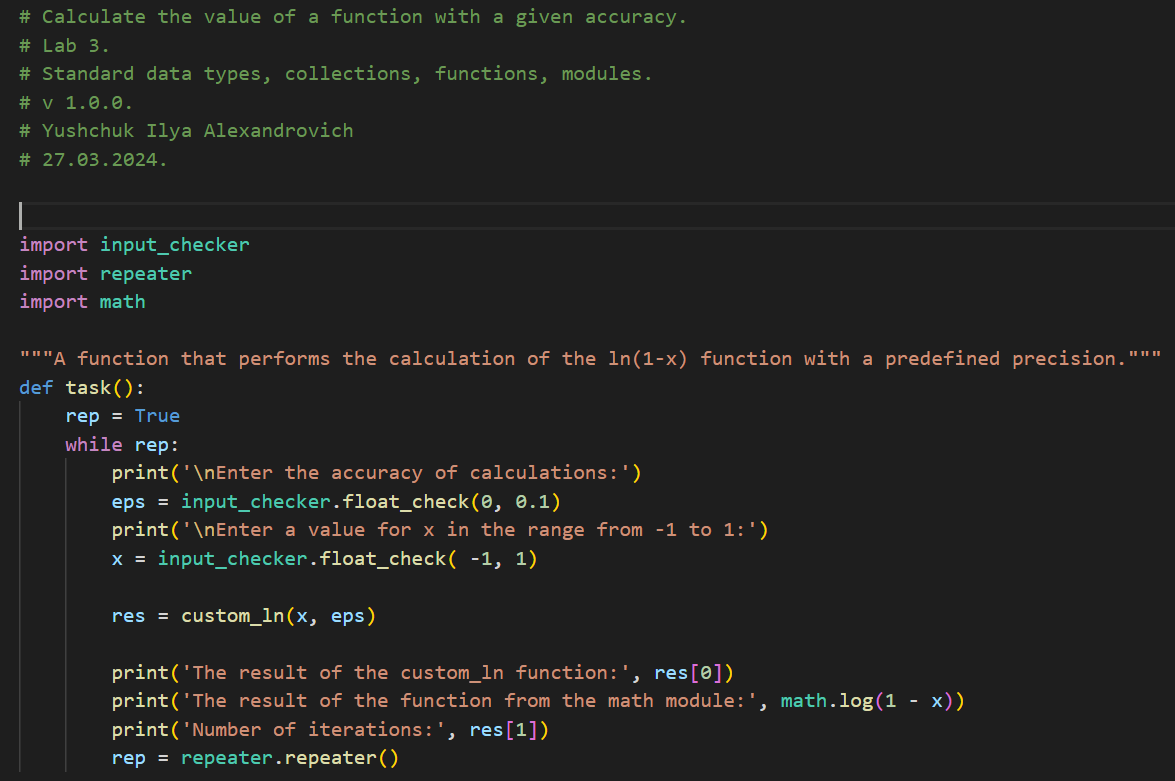
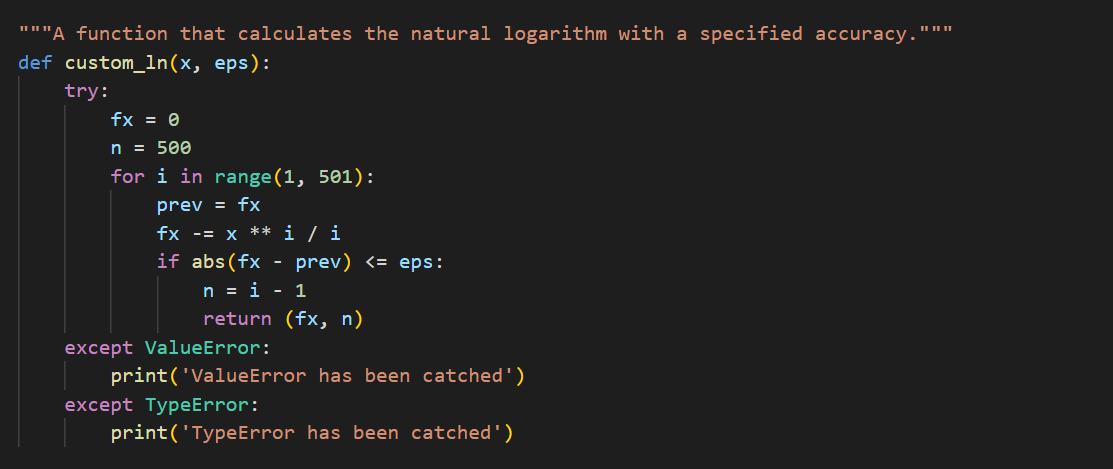
Вывести количество членов ряда, необходимых для достижения указанной точности вычислений. Результат получить в виде:



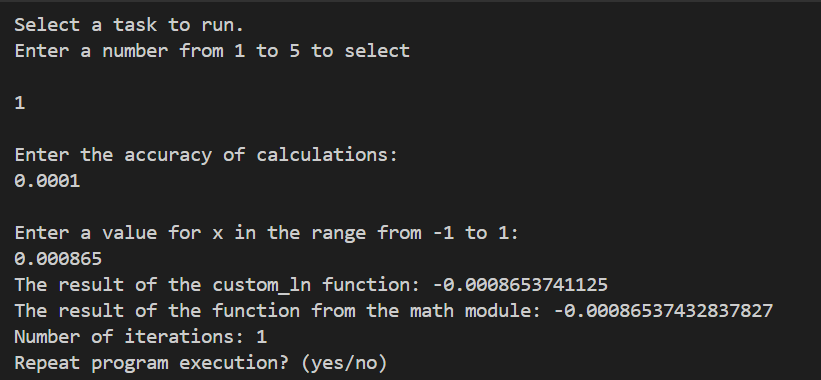
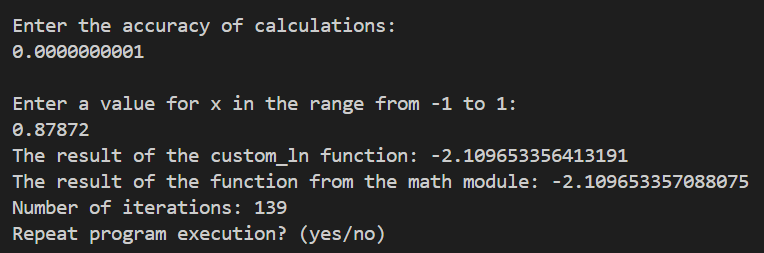
Здесь x – значение аргумента, F(x) – значение функции, n – количество просуммированных членов ряда, Math F(x) – значение функции, вычисленное с помощью модуля math.x

|  |  |
| --- | --- |
| **Вар-т** | **Условие** |
| 29. |  |

**Код программы**

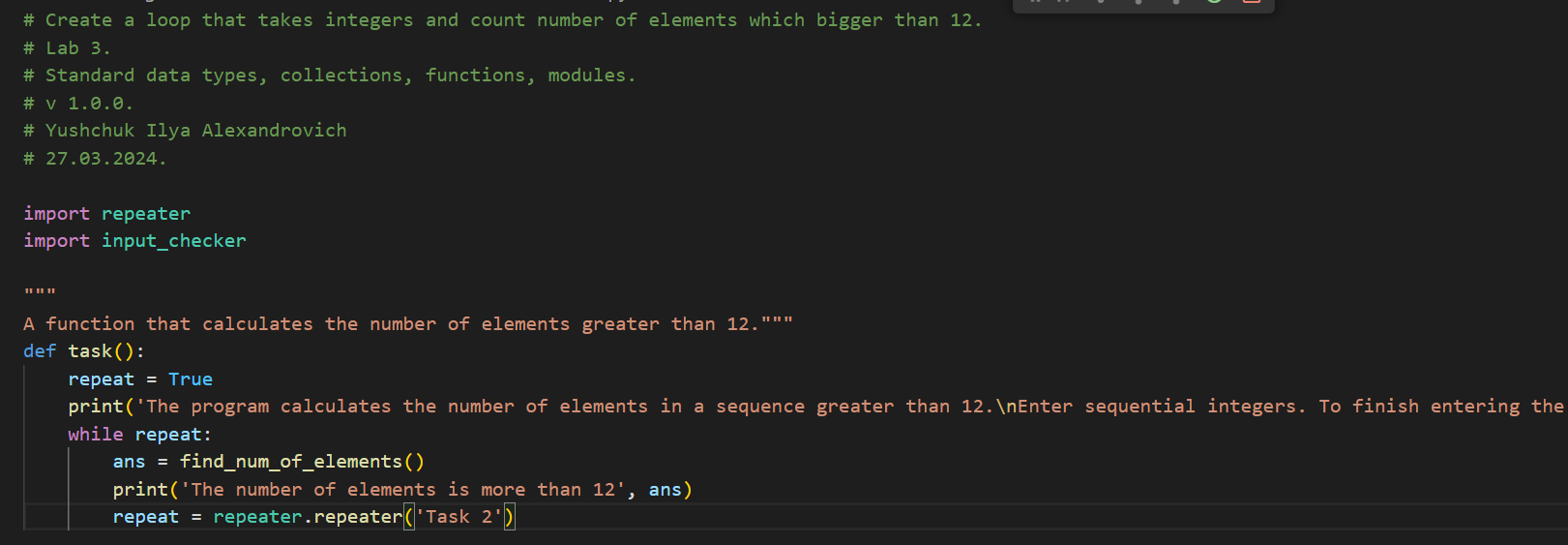
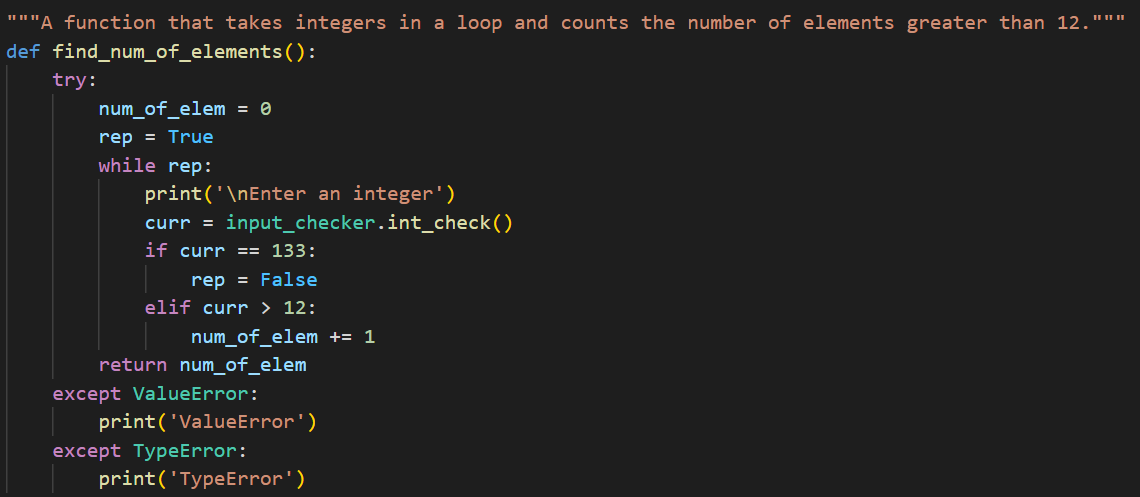
**** ****

**Результат работы программы**

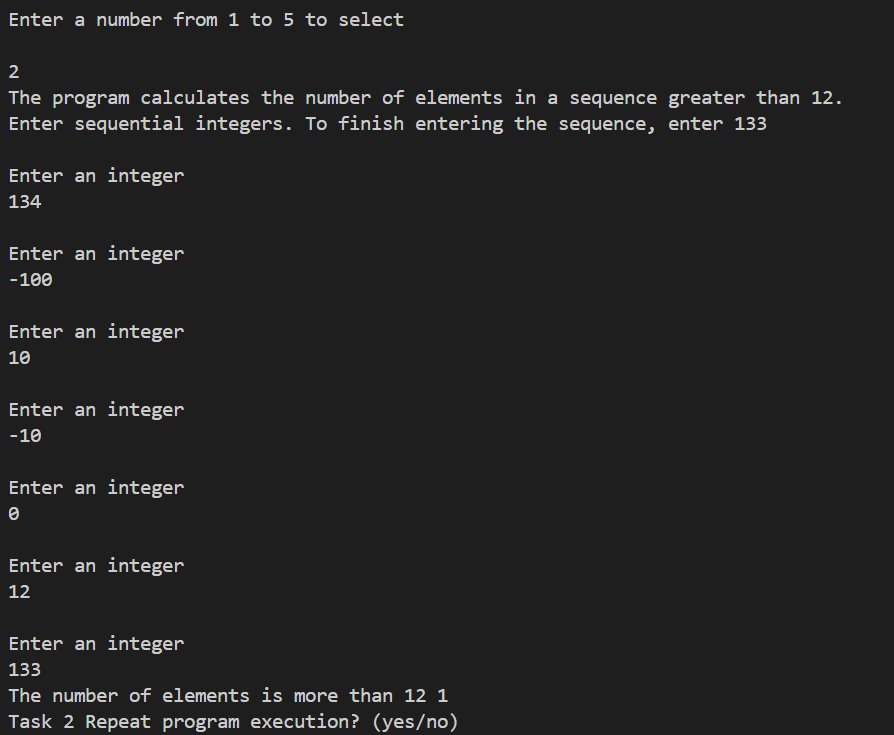
**** ****

**Задание 2.** В соответствии с заданием своего варианта составить программу для нахождения суммы последовательности чисел.

|  |  |
| --- | --- |
| Вар-т | Условие |
| 29 | Организовать цикл, который принимает целые числа и вычисляет количество чисел, больше 12. Окончание цикла – ввод 133 |

**Код программы**  

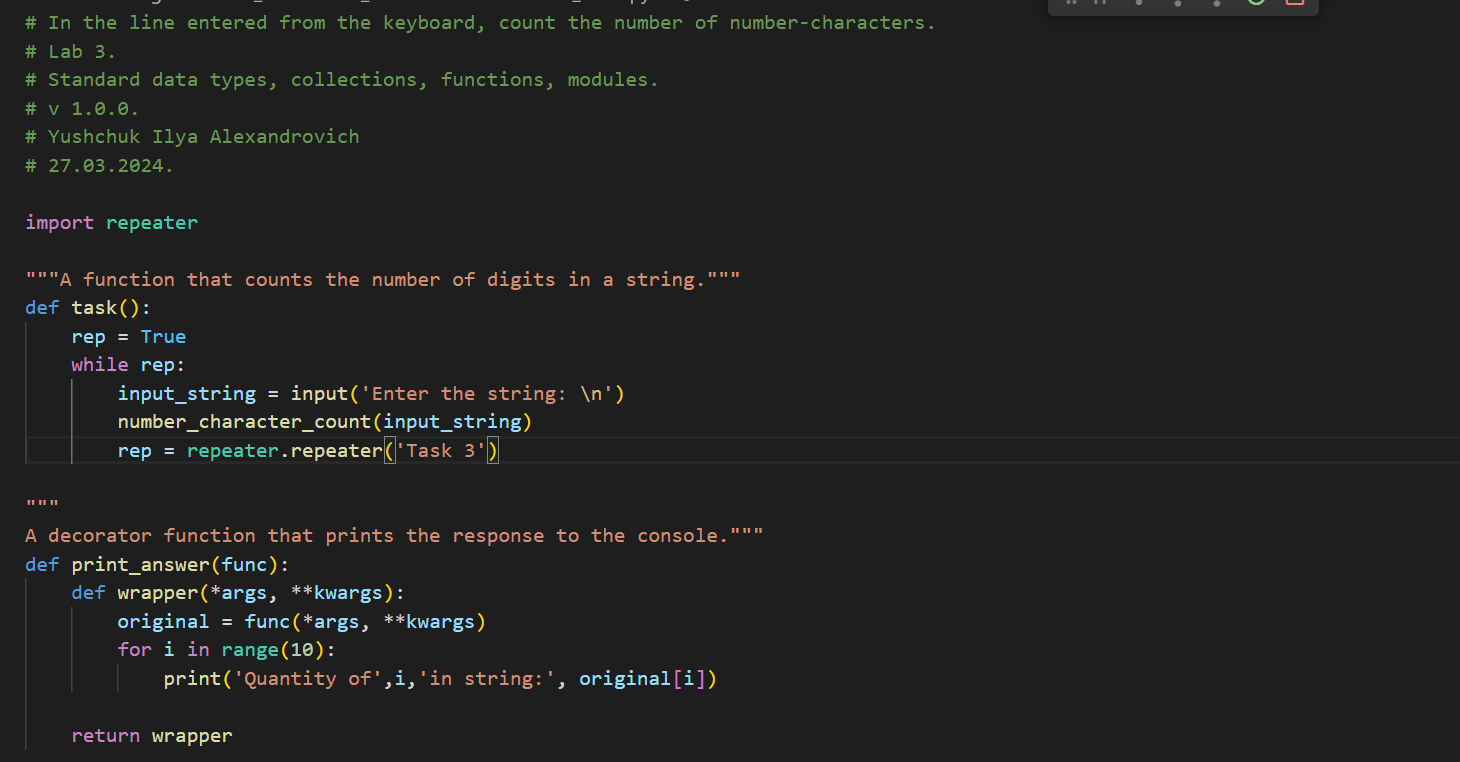
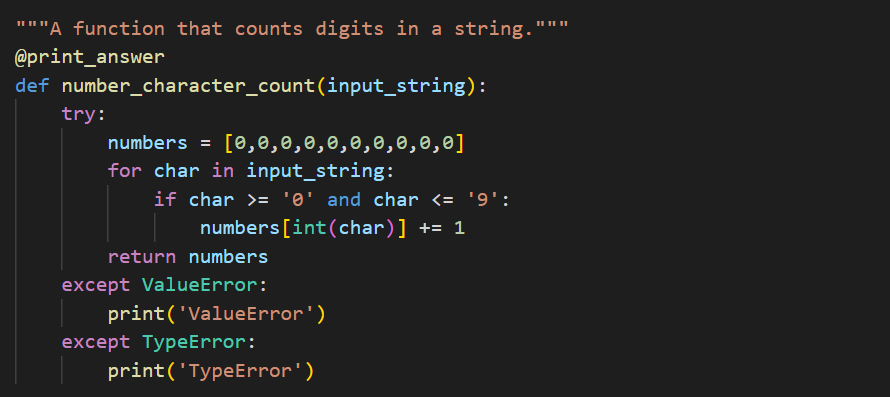
**Результат работы программы**

****

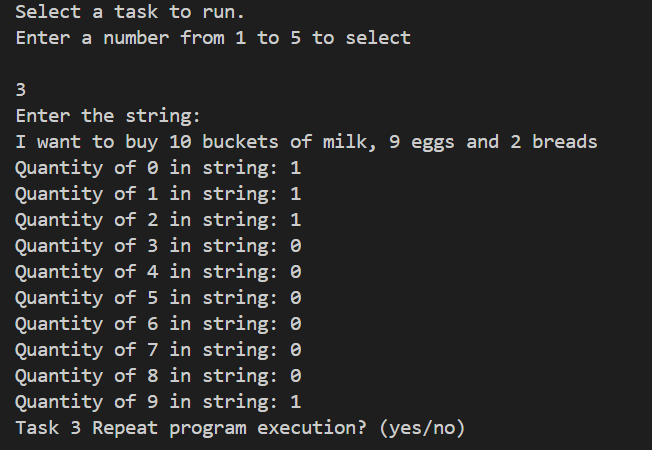
**Задание 3.** **Не использовать регулярные выражения**. В соответствии с заданием своего варианта составить программу для анализа текста, вводимого с клавиатуры.

|  |  |
| --- | --- |
| Вар-т | Условие |
| 29 | В строке, вводимой с клавиатуры, подсчитать количество цифр. |

**Код программы**

**** ****

**Результат работы программы**



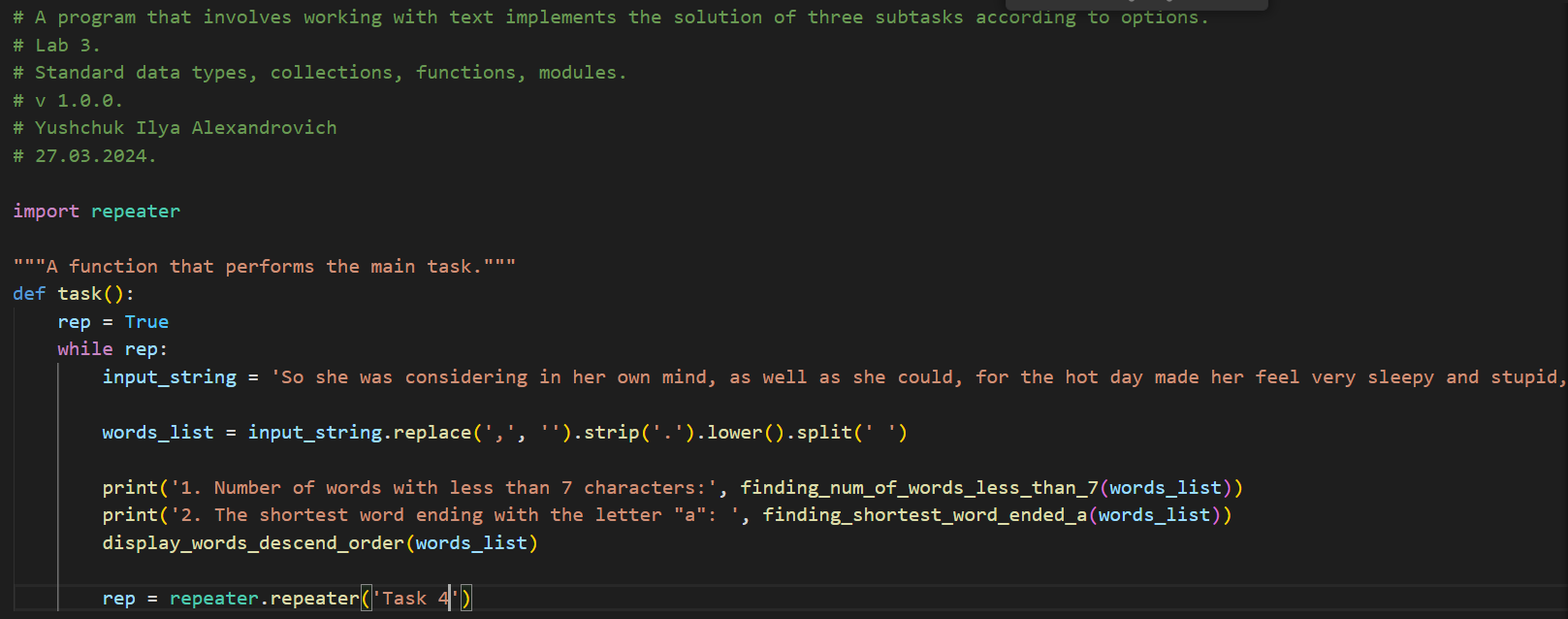
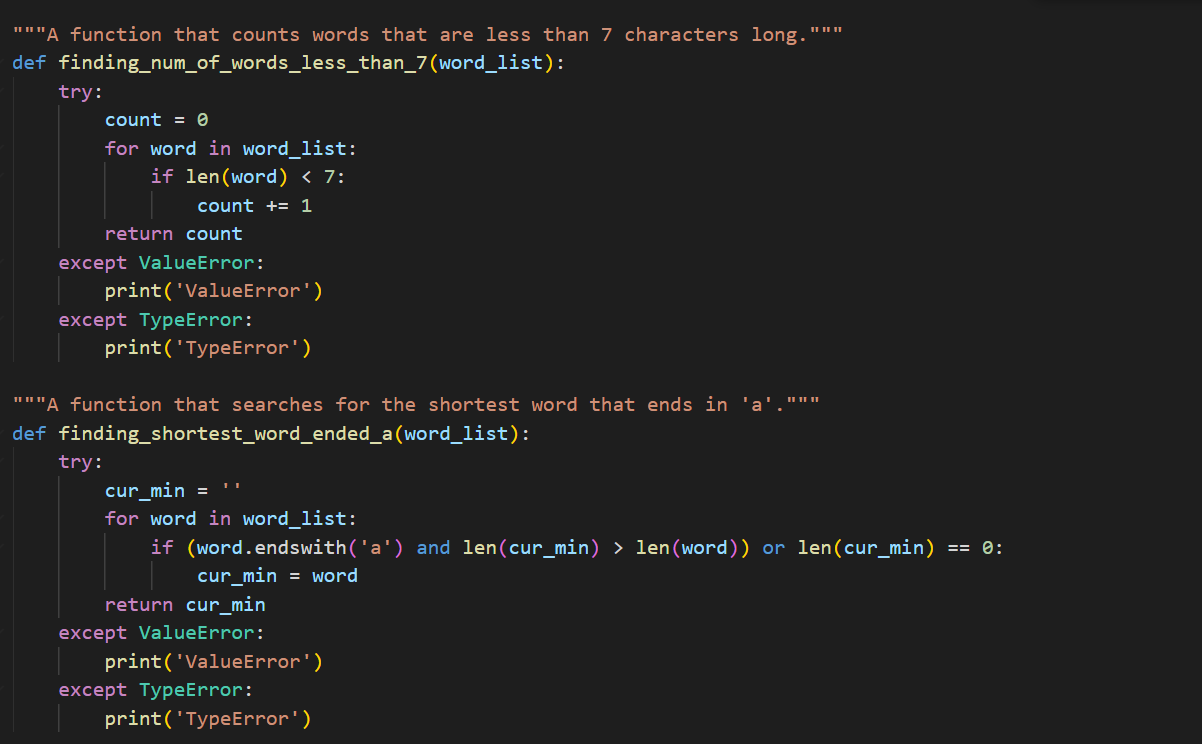
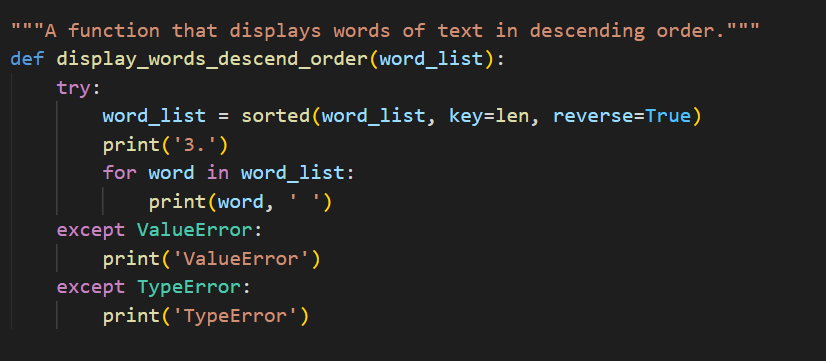
**Задание 4. Не использовать регулярные выражения**. Дана строка текста, в которой слова разделены пробелами и запятыми. В соответствии с заданием своего варианта составьте программу для анализа строки, инициализированной в коде программы:

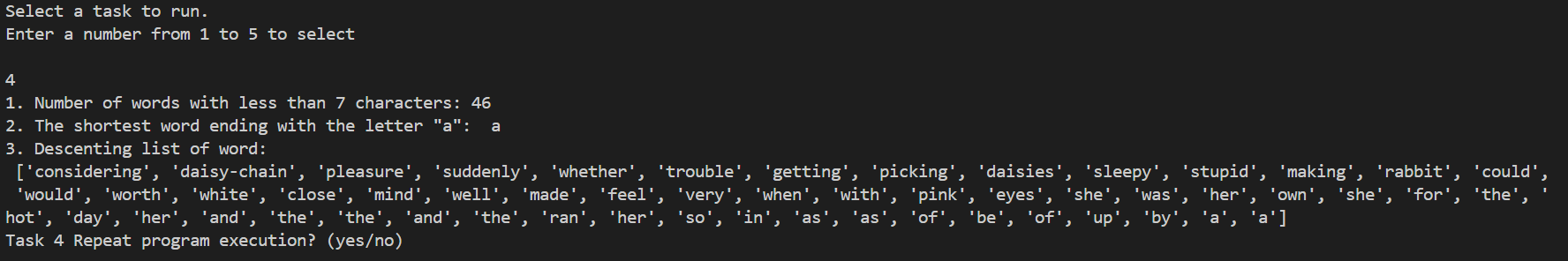
«So she was considering in her own mind, as well as she could, for the hot day made her feel very sleepy and stupid, whether the pleasure of making a daisy-chain would be worth the trouble of getting up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.»

Если не оговорено иное, то регистр букв при решении задачи не имеет значения.

|  |  |
| --- | --- |
| Вар-т | Условие |
| 29 | а) определить число слов, длина которых меньше 7 символов;  б) найти самое короткое слово, заканчивающееся на букву 'a';  в) вывести все слова в порядке убывания их длин |

**Код программы**

**** **** 

**Результат работы программы** 

**Задание 5.** В соответствии с заданием своего варианта составить программу для обработки вещественных списков. Программа должна содержать следующие базовые функции:

1) ввод элементов списка пользователем;

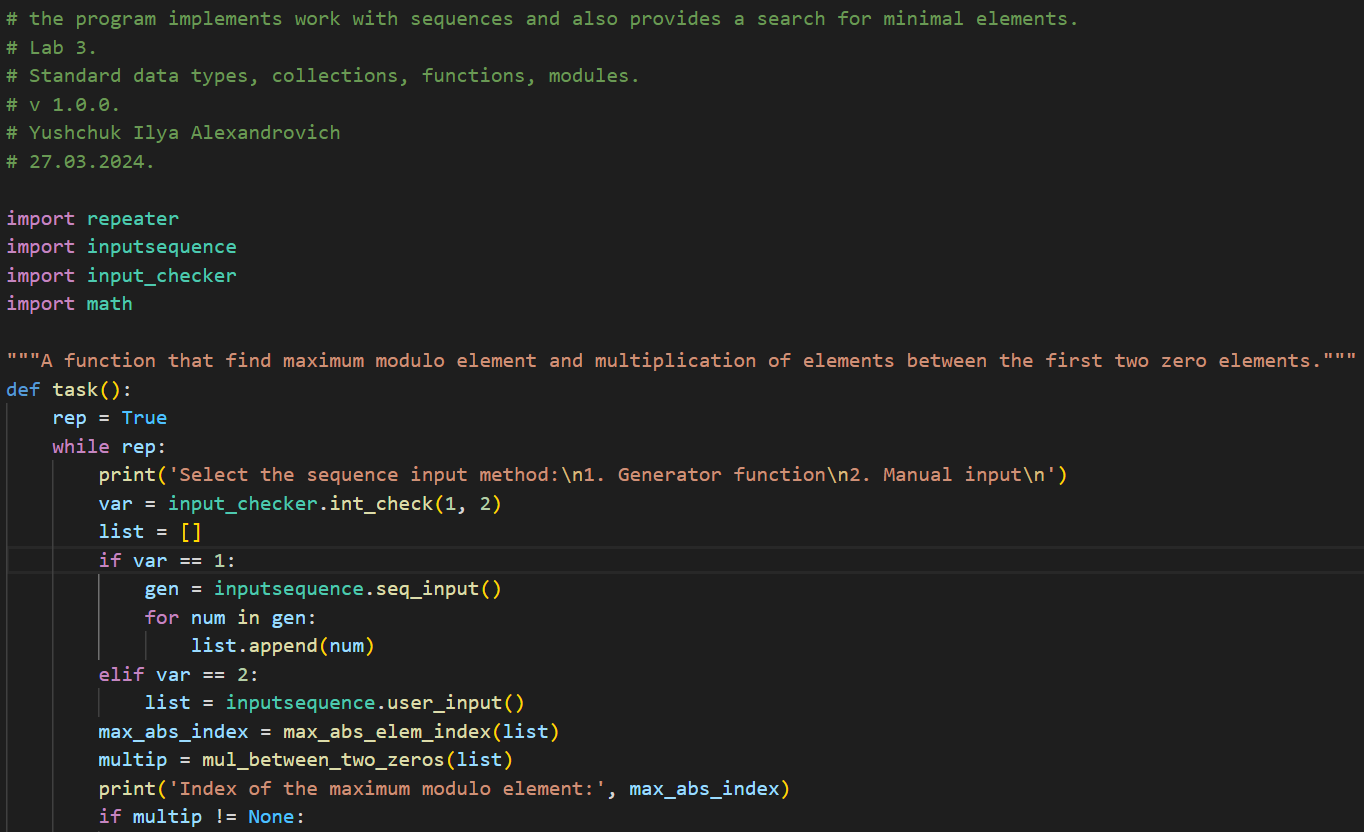
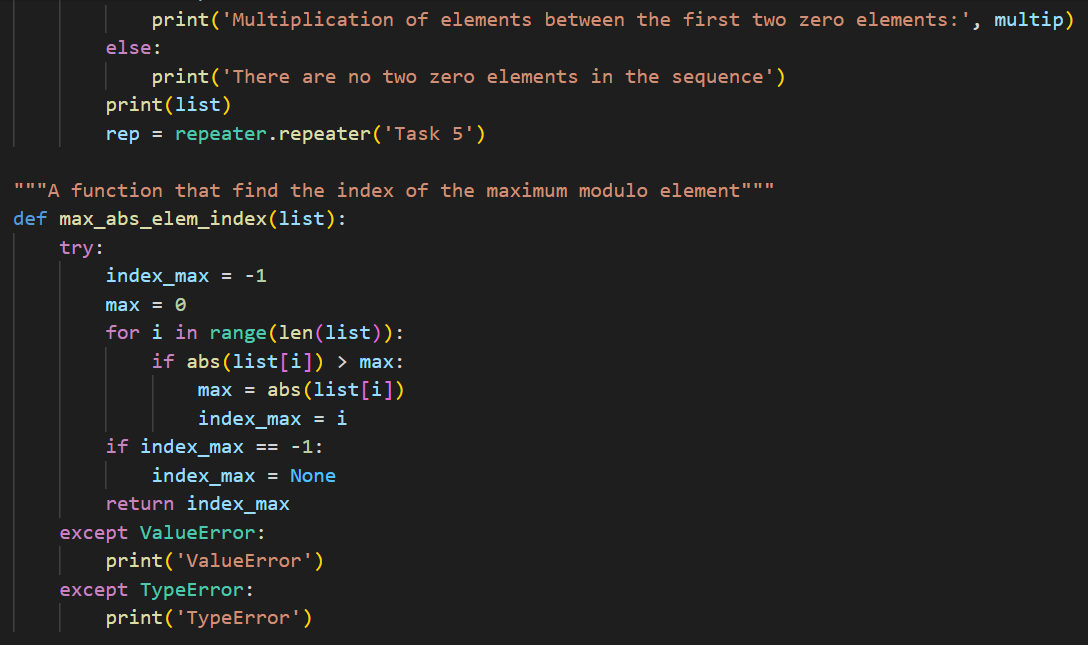
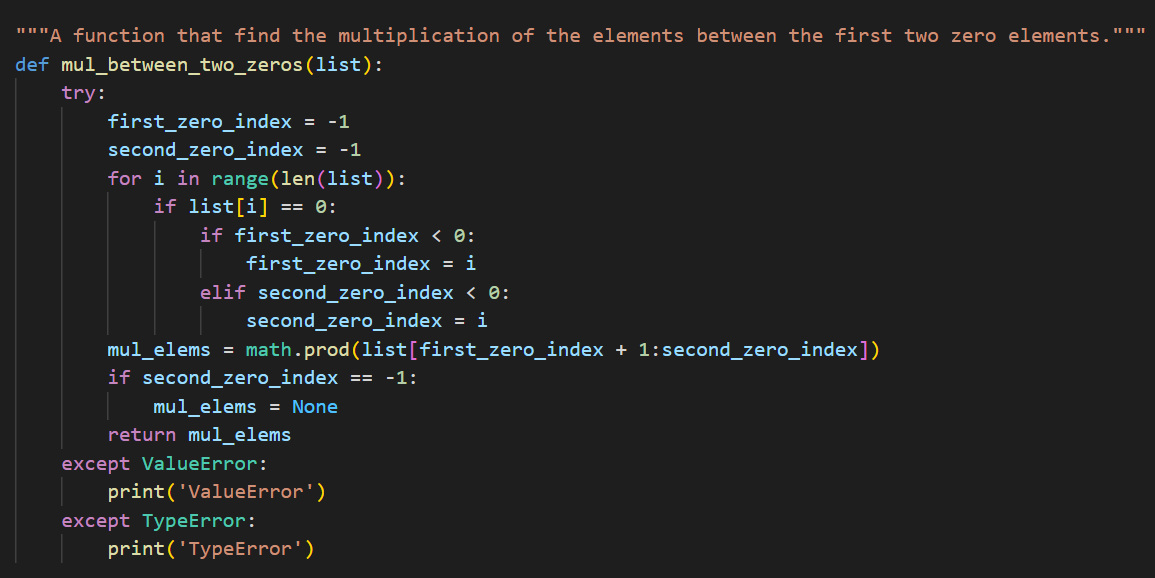
2) проверка корректности вводимых данных;

3) реализация основного задания с выводом результатов;

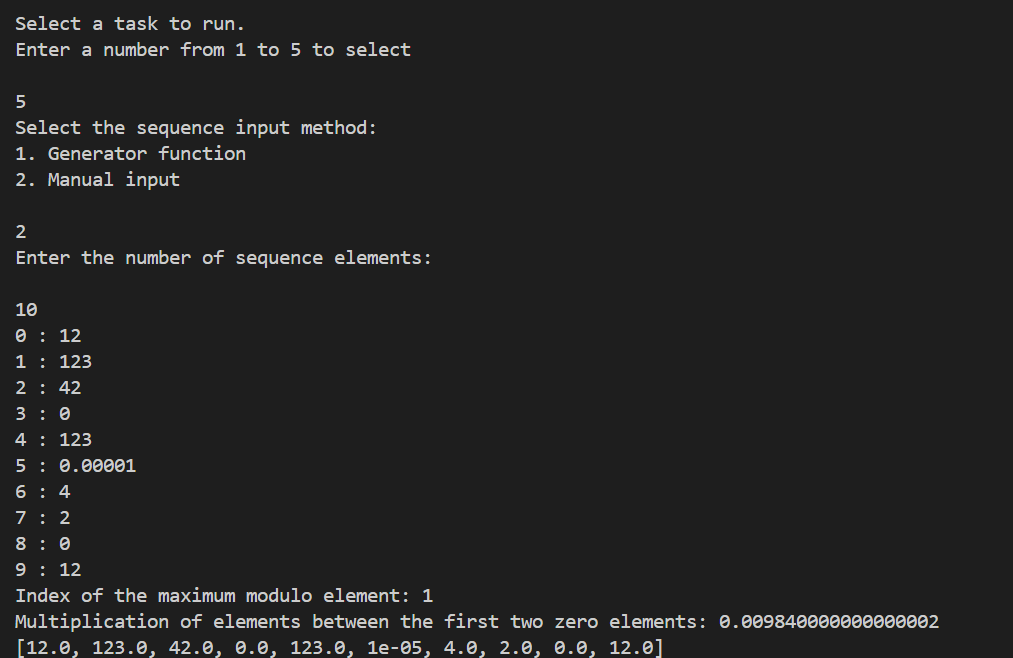
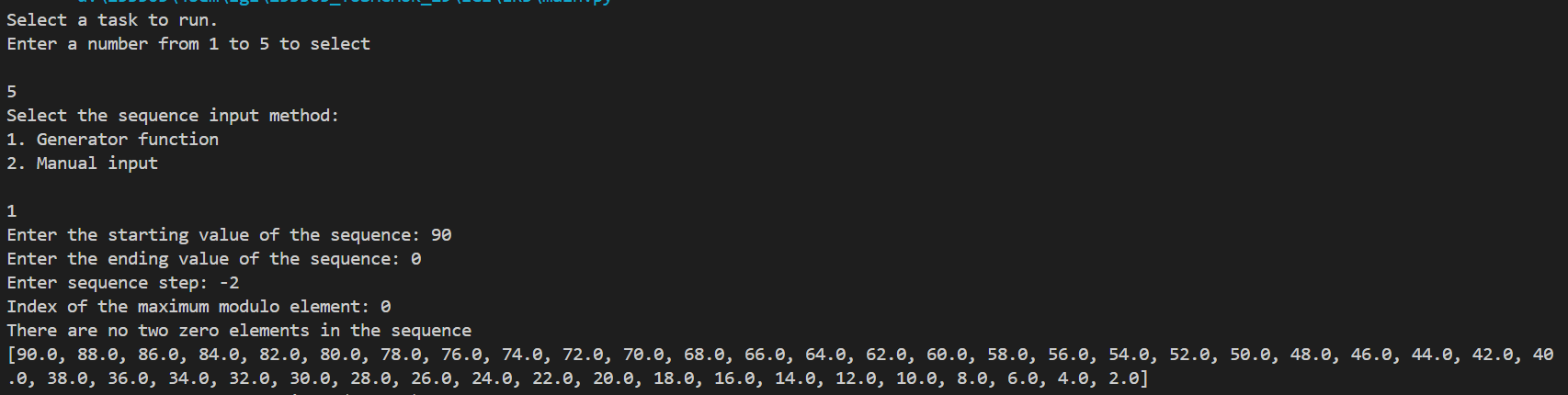
4) вывод списка на экран.

|  |  |
| --- | --- |
| Вар | Условие |
| 4 | Найти номер максимального по модулю элемента списка и произведение элементов, расположенных между первым и вторым нулевыми элементами |

**Код программы**

**** **** 

**Результат работы программы**

**Контрольные вопросы**

1. Перечислите основные управляющие конструкции.
2. Как в языке Python реализуется механизм истинности-ложности? Может ли значение быть условием?
3. С помощью каких операторов можно комбинировать в одной условной конструкции if несколько условий? Какой механизм оптимизации применяет интерпретатор Python для эффективного вычисления результата комбинированных условных выражений?
4. Опишите синтаксис условной конструкции if-else. Представьте примерную блок-схему конструкции.
5. Опишите синтаксис условной конструкции elif. Представьте примерную блок-схему конструкции.
6. Чем использование elif будет отличаться от использования вложенных условных конструкций if-else?
7. Для чего используются циклы? Что такое итерация?
8. Какие разновидности циклов существуют?
9. Описать Python-синтаксис цикла с предусловием while.
10. Какова роль оператора break в теле цикла?
11. Какова роль оператора continue в теле цикла?
12. Какова роль оператора pass в теле цикла?
13. Может ли выражение после ключевого слова while содержать истинное значение или значение других типов данных?
14. Что такое бесконечный цикл? Когда он применяется? Привести пример кода организации диалога на тему завершения программы, либо повторного выполнения программы.
15. Если необходимо использовать вложенные циклы while для вывода элементов прямоугольной матрицы в виде строк и столбцов, какой из циклов будет печатать строки: внутренний или внешний?
16. Что такое функция? Как описывается функция в Python?
17. Зачем нужны функции?
18. Для чего используется оператор return в функциях? Как возвратить из функции несколько значений?
19. Чем формальные параметры отличаются от фактических?
20. Что такое позиционные параметры?
21. Что такое параметры по умолчанию?
22. Чем отличается глобальная переменная от локальной?
23. Чем характеризуется строковый тип данных в Python?
24. Какие есть способы объявления строк в Python?
25. Что такое неизменяемость строк?
26. Зачем нужна индексация строк, и как ее использовать?
27. Зачем и как используются срезы строк?
28. Какие основные операторы используются для работы со строками?
29. Какие основные встроенные функции класса str используются для работы со строками?
30. Приведите примеры объявления каждого из высокоуровневых
31. типов данных.
32. Объясните понятие «распаковка последовательности».
33. Какое главное отличие списков от кортежей? Когда лучше использовать кортежи, а когда – списки?
34. Как получить доступ к элементам словаря? Можно ли использовать индексацию для словарей?
35. Зачем и как используются срезы?
36. Какие операторы и встроенные функции используются для работы с кортежами, списками, словарями и множествами?
37. Какие методы есть у каждого из классов, отвечающих за каждый высокоуровневый тип данных (tuple, list, dict и set)? Опишите наиболее востребованные.