**Практическая работа №7. Функции. Область видимости переменных**

Задание

Выбрать задачи из файлов с вариантами в соответствии с таблицей.

Ссылка на варианты [ElenaKuzmina/PR7\_Function](https://github.com/ElenaKuzmina/PR7_Function)

Написать программный модуль, содержащий функции, реализующие решение задачи. Обязательно должна быть функция main(), содержащая вызовы остальных функций. В отчет включить для каждой задачи:

1. ФИО студента, группа
2. Постановка задачи
3. Описание переменных с указанием имени, типа, назначения, уровня видимости (в виде таблицы)
4. Словесный алгоритм решения задачи
5. Программный код модуля
6. Скриншоты ошибок с описанием ошибки и способом ее исправления
7. Скриншоты тестирования программного модуля задачи
8. Ссылка на свой репозиторий github с программным модулем

Таблица вариантов

|  |  |  |
| --- | --- | --- |
| Студент | Задача 1 | Задача 2 |
| [Соловьева Юлия](https://school.mosreg.ru/marks?school=2000002775450&group=2257826504610747540&student=2000000669920&tab=stats) | Вариант 3, задача 1 | Вариант 4, задача 2 |

**Соловьева Юлия ИСП 23.1А**

**Задача 1**

**Вариант 3, задача 1**

**Формат ввода**

Вводится строка слов, записанных через двоеточие и пробел, затем вводится число.

**Формат вывода**

Через запятую и пробел, с большой буквы, остальные маленькие, в порядке ввода вывести слова с длиной, большей числа, но ему не кратной.

**Пример 1**

| Ввод | Вывод |
| --- | --- |
| Most: of: the: people: living: on: the: planet: did: not: find: happiness: in: life  3 | Most, Find, Life |

**Пример 2**

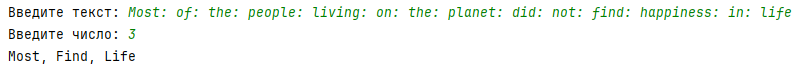
| Ввод | Вывод |
| --- | --- |
| there: was: nothing: to: mess: with: the: trees: either  2 | There, Was, Nothing, The, Trees |

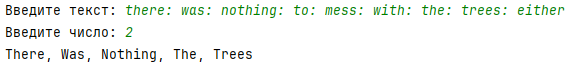
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Идентификатор | Тип | Назначение | Уровень видимости |
| 1 | string | str | Хранит строку, введённую пользователем. | Глобальная |
| 2 | n | int | Хранит целое число, введённое пользователем. | Глобальная |
| 3 | words | list | Список слов, полученных путём разделения строки string по символу ": ". | Локальная |
| 4 | filter | list | Список, содержащий отфильтрованные слова из списка words, которые имеют длину больше n и не являются кратными n. | Локальная |
| 5 | output\_string | str | Хранит строку, состоящую из отфильтрованных слов, объединённых запятой и пробелом. | Локальная |

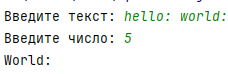
**Программа:**

string = input('Введите текст: ')  
n = int(input('Введите число: '))  
def main():  
  
 words = string.split(': ')  
  
 filter = [  
 word.capitalize()  
 for word in words  
 if len(word) > n and len(word) % n != 0  
]  
  
 output\_string = ', '.join(filter)  
 print(output\_string)  
main()

**Тест:**

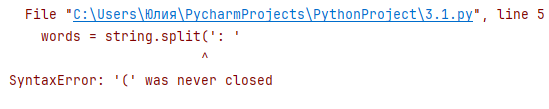




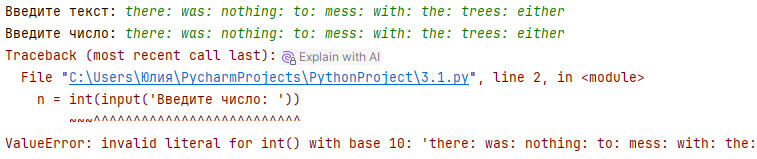


**Ошибки:**

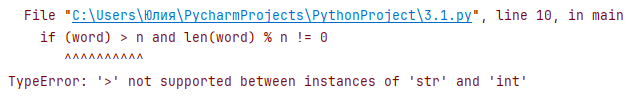
1. Не поставлена скобка. Чтобы исправить ошибку, нужно поставить недостающую скобку.

****

1. Неправильный ввод. Чтобы исправить ошибку, нужно ввести число.



1. Неправильный тип данных. Чтобы исправить ошибку, нужно поставить len.



**Задача 2**

**Вариант 4, задача 2**

**Формат ввода**

Вводится три строки чисел, записанных через запятую и пробел.

**Формат вывода**

В одну строку через пробел запишите числа, которые были во всех трех строках. Выводить без повторений в любом порядке.

Затем выведите наибольшее из таких чисел

**Пример 1**

| Ввод | Вывод |
| --- | --- |
| 20, 1, 2, 16, 1, 16, 20, 11  6, 3, 1, 4, 2, 2, 8, 3  9, 9, 13, 2, 10, 20, 1 | 1 2  2 |

**Пример 2**

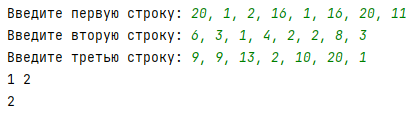
| Ввод | Вывод |
| --- | --- |
| 4, 16, 10, 8, 2, 14, 3, 6  19, 6, 3, 17, 1, 8, 2, 6  6, 3, 8, 3, 8, 11, 2 | 8 2 3 6  8 |

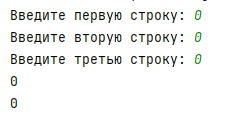
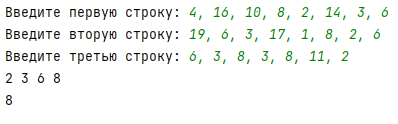
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Идентификатор | Тип | Назначение | Уровень видимости |
| 1 | n1 | str | Хранит первую строку, введённую пользователем. | Глобальная |
| 2 | n2 | str | Хранит вторую строку, введённую пользователем. | Глобальная |
| 3 | n3 | str | Хранит третью строку, введённую пользователем. | Глобальная |
| 4 | set1 | set | Хранит множество целых чисел, преобразованных из первой строки. | Локальная |
| 5 | set2 | set | Хранит множество целых чисел, преобразованных из второй строки. | Локальная |
| 6 | set3 | set | Хранит множество целых чисел, преобразованных из третьей строки. | Локальная |
| 7 | peresechenie | set | Содержит пересечение трёх множеств set1, set2 и set3. | Локальная |
| 8 | list\_num | list | Список, содержащий отсортированные целые числа из пересечения. | Локальная |

Программа:

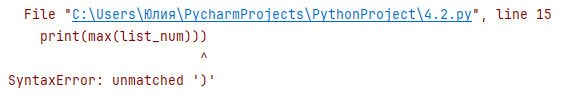
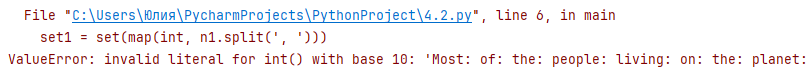
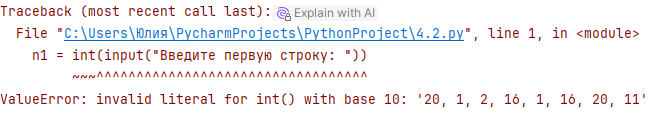
n1 = input("Введите первую строку: ")  
n2 = input("Введите вторую строку: ")  
n3 = input("Введите третью строку: ")  
def main():  
  
 set1 = set(map(int, n1.split(', ')))  
 set2 = set(map(int, n2.split(', ')))  
 set3 = set(map(int, n3.split(', ')))  
  
 peresechenie = set1.intersection(set2).intersection(set3)  
  
 if peresechenie:  
 list\_num = sorted(peresechenie)  
 print(" ".join(map(str, list\_num)))  
 print(max(list\_num))  
 else:  
 print("Нет общих чисел")  
  
main()

Тест:





**Ошибки:**

1. Поставлена лишняя скобка. Чтобы исправить ошибку, нужно убрать ненужную скобку. ****
2. Неверный ввод. Чтобы исправить ошибку вместо строки, нужно ввести число.****
3. Неверный тип данных. Чтобы исправить ошибку нужно убрать int.****

Ссылка на гитхаб:

https://github.com/SolovevaYuliya/OAP-PR