**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)

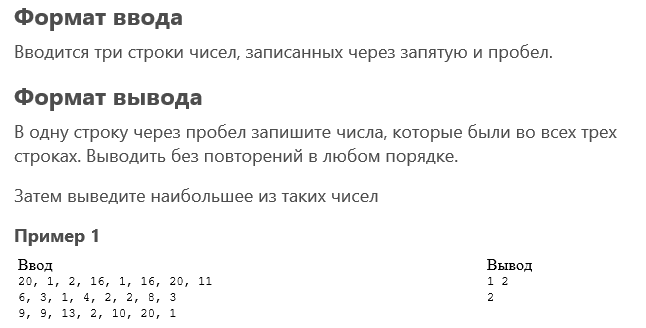
Написать программный модуль, содержащий функции, реализующие решение задачи. Обязательно должна быть функция main(), содержащая вызовы остальных функций. В отчет включить для каждой задачи:

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 |
| [Колпаков Матвей](https://school.mosreg.ru/marks?school=2000002775450&group=2257826504610747540&student=1000007761619&tab=stats) | Вариант 4, задача 2 | Вариант 4, задача 3 |

Задача №1

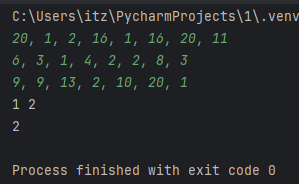
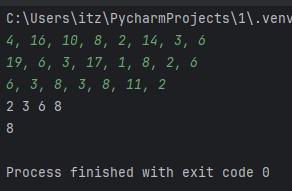
1. Колпаков Матвей Николаевич ИСП.23.1А
2. 

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Тип данных | Назначение | Уровень видимости |
| Input\_str1 | String | Ввод строки | Локальный |
| Input\_str2 | String | Ввод строки | Локальный |
| Input\_str3 | String | Ввод строки | Локальный |
| Set1 | Set | Преобразование в множество первой строки | Локальный |
| Set2 | Set | Преобразование в множество второй строки | Локальный |
| Set3 | Set | Преобразование в множество третей строки | Локальный |
| Common\_numbers | Set(int) | Множество чисел, встречающихся во всех трех строчках | Локальный |
| Max\_common | int | Наибольшее число | Локальный |

1. Вводим три входные строки, функции преобразовывают строку в множества, убирая дубликаты, ищем пересечение этих множеств, выводим через пробел с помощью метода join() и выводятся дубликаты цифр всех трех строк и максимальное число.

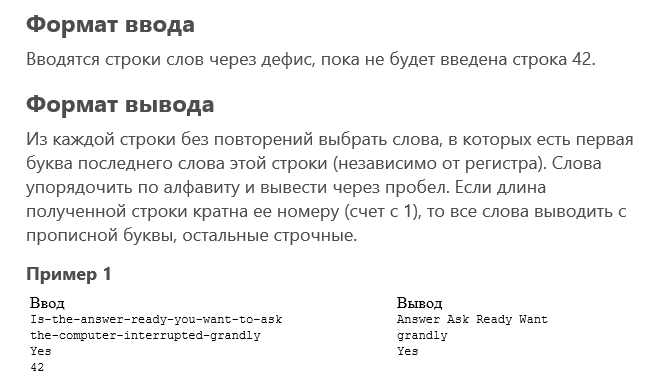
5) def inputt(inputq):  
 return set(map(int, inputq.split(', ')))  
  
  
def find\_common\_numbers(set1, set2, set3):  
 return set1 & set2 & set3  
  
  
def print\_results(common\_numbers):  
 if common\_numbers:  
 print(" ".join(map(str, sorted(common\_numbers))))  
 print(max(common\_numbers))  
 else:  
 print()  
  
  
  
def main():  
 input\_str1 = input().strip()  
 input\_str2 = input().strip()  
 input\_str3 = input().strip()  
  
 set1 = inputt(input\_str1)  
 set2 = inputt(input\_str2)  
 set3 = inputt(input\_str3)  
  
 common\_numbers = find\_common\_numbers(set1, set2, set3)  
 print\_results(common\_numbers)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

6)

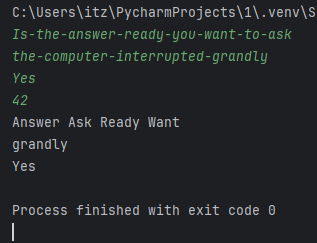
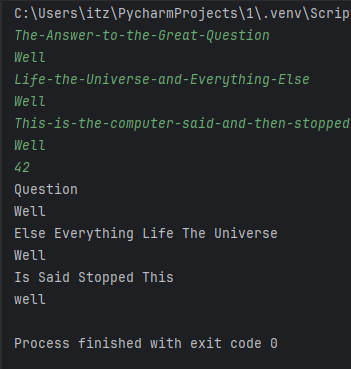
7)  

8) https://github.com/1tziced/7pr

Задача №2

1. Колпаков Матвей Николаевич ИСП.23.1А
2. 

|  |  |  |  |
| --- | --- | --- | --- |
| Имя | Тип данных | Назначение | Уровень видимости |
| Line | str | Введенная строка | Локальный |
| Words | List(str) | Список слов, полученных из строки | Локальный |
| Last\_words | Str | Последнее слово в строке | Локальный |
| First\_letter | Str | Первая буква последнего слова | Локальный |
| Filtered\_words | Str | Уникальные слова, содержащие fisrt\_letter | Локальный |
| Sorted\_words | List(str) | Отсортированные отфильтрованные слова | Локальный |
| Result | Str | Итоговая строка для вывода | Локальный |
| Line\_number | int | Номер текущей строки | Локальный |

1. Считываем строки,пока не будет введена строка «42», разделяем строку на слова, используя дефис в качестве разделителя, определяем последнее слово и первую букву последнего слова, выбрать слова и удалить повторения, отстортировывание слов, определение длины строки и проверка кратности. Повторение процесса для каждой новой строки пока не будет введено «42».
2. def process\_line(line, line\_number):  
    words = line.split('-')  
    last\_word = words[-1]  
    first\_letter = last\_word[0].lower()  
     
    filtered\_words = {word for word in words if first\_letter in word.lower()}  
    sorted\_words = sorted(filtered\_words, key=str.lower)  
    result = " ".join(sorted\_words)  
     
    if len(result) % line\_number == 0:  
    result = result.title()  
    else:  
    result = result.lower()  
     
    print(result)  
     
     
     
   def main():  
    line\_number = 1  
    while True:  
    line = input().strip()  
    if line == "42":  
    break  
    process\_line(line, line\_number)  
    line\_number += 1  
     
     
   if \_\_name\_\_ == "\_\_main\_\_":  
    main()
3. 
4. https://github.com/1tziced/7pr