**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 |
| [Хапов Дмитрий](https://school.mosreg.ru/marks?school=2000002775450&group=2257826504610747540&student=2000000156975&tab=stats) | Вариант 2, задача 1 | Вариант 1, задача 2 |

**Задача 1**

**Формат ввода**

Вводится число и строка слов, записанных через точку с запятой и пробел.

**Формат вывода**

Через дефис, окружённый пробелами, выведите в порядке ввода слова с длиной, не кратной числу. Слова вывести большими буквами, чтобы стало ясно – вот оно, счастье!

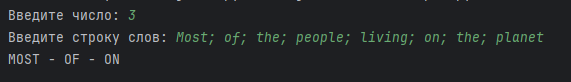
**Пример 1**

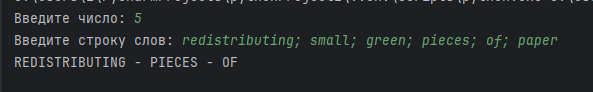
| Ввод | Вывод |
| --- | --- |
| 3  Most; of; the; people; living; on; the; planet | MOST - OF - ON |

**Пример 2**

| Ввод | Вывод |
| --- | --- |
| 5  redistributing; small; green; pieces; of; paper | REDISTRIBUTING - PIECES - OF |

def main(number, words\_string):  
 # Разбиваем строку на отдельные слова  
 words = words\_string.split('; ')  
 # Фильтруем слова, длина которых не кратна числу  
 filtered\_words = [word.upper() for word in words if len(word) % number != 0]  
 # Формируем итоговую строку с разделением через ' - '  
 result = ' - '.join(filtered\_words)  
 return result  
  
  
number = int(input("Введите число: "))  
string = input("Введите строку слов: ")  
print(main(number, string))





**Задача 2**

**Формат ввода**

Вводится три строки чисел, разделённых пробелами.

**Формат вывода**

В одну строку через точку с запятой и пробел вывести числа, которые есть во второй и третьей строке, но которых нет в первой. Выводить без повторений в любом порядке.

Затем вывести сумму выбранных чисел.

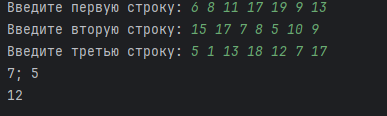
**Пример 1**

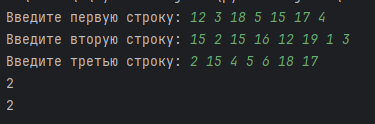
| Ввод | Вывод |
| --- | --- |
| 6 8 11 17 19 9 13  15 17 7 8 5 10 9  5 1 13 18 12 7 17 | 5; 7  12 |

**Пример 2**

| Ввод | Вывод |
| --- | --- |
| 12 3 18 5 15 17 4  15 2 15 16 12 19 1 3  2 15 4 5 6 18 17 | 2  2 |

def main():  
 # 1. Считываем три строки чисел  
 first\_line = input("Введите первую строку: ")  
 second\_line = input("Введите вторую строку: ")  
 third\_line = input("Введите третью строку: ")  
  
 # 2. Преобразуем строки в списки  
 first\_nums = first\_line.split()  
 second\_nums = second\_line.split()  
 third\_nums = third\_line.split()  
  
 # 3. Создаём множество из первой строки  
 first\_set = set(first\_nums)  
  
 # 4. Находим пересечение во второй и третьей строках  
 common\_numbers = set(second\_nums) & set(third\_nums)  
  
 # 5. Фильтруем, удаляя числа, которые есть в первой строке  
 result\_set = {num for num in common\_numbers if num not in first\_set}  
  
 # 6. Форматируем вывод  
 result\_output = "; ".join(result\_set)  
  
 # 7. Подсчитываем сумму  
 result\_sum = sum(map(int, result\_set))  
  
 # 8. Выводим результаты  
 print(result\_output)  
 print(result\_sum)  
  
  
main()





<https://github.com/Xedos808/dz>