МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет по лабораторной работе №13 «Функции с переменным числом параметров в Python»

по дисциплине «Основы программной инженерии»

Выполнила:
Первых Дарья Александровна,
2 курс, группа ПИЖ-б-о-20-1,
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г.

ВЫПОЛНЕНИЕ

```
def print_these(a, b, c):
    print(a, "is stored in a")
    print(b, "is stored in b")
    print(c, "is stored in c")

print_these(1, 2, 3)
```

```
print_these()
main
C:\Users\podar\study\anaconda\envs\13
1 is stored in a
2 is stored in b
3 is stored in c
```

Рисунок 1 – Пример простейшей функции сопоставления позиции аргументов и параметров

```
def print_these(a, b, c):
    print(a, "is stored in a")
    print(b, "is stored in b")
    print(c, "is stored in c")

print_these(1, 2)
```

```
main
C:\Users\podar\study\anaconda\envs\13LR\python.exe "C:/Users/podar/st
Traceback (most recent call last):
  File "C:\Users\podar\study\PyCharm Community Edition 2021.2.3\13LR\
    print_these(1, 2)
TypeError: print_these() missing 1 required positional argument: 'c'
```

Рисунок 2 – Результат неверного вывода команд

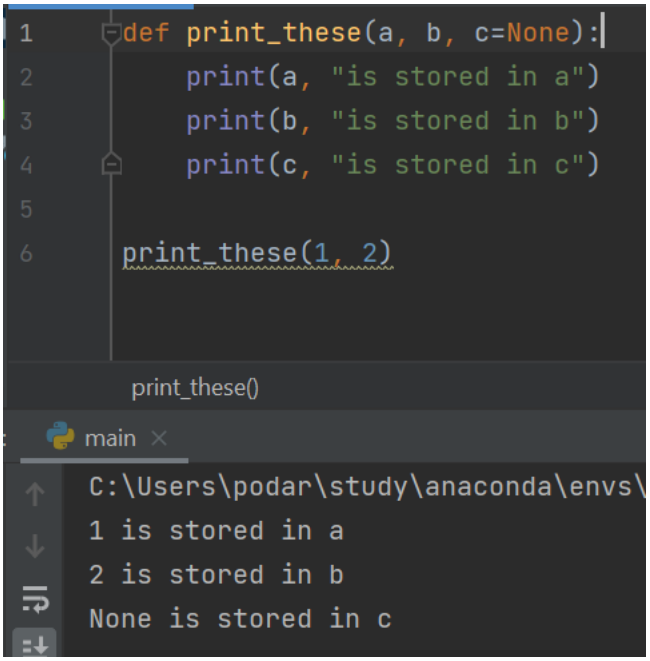

Рисунок 3 – Применение опциональных параметров

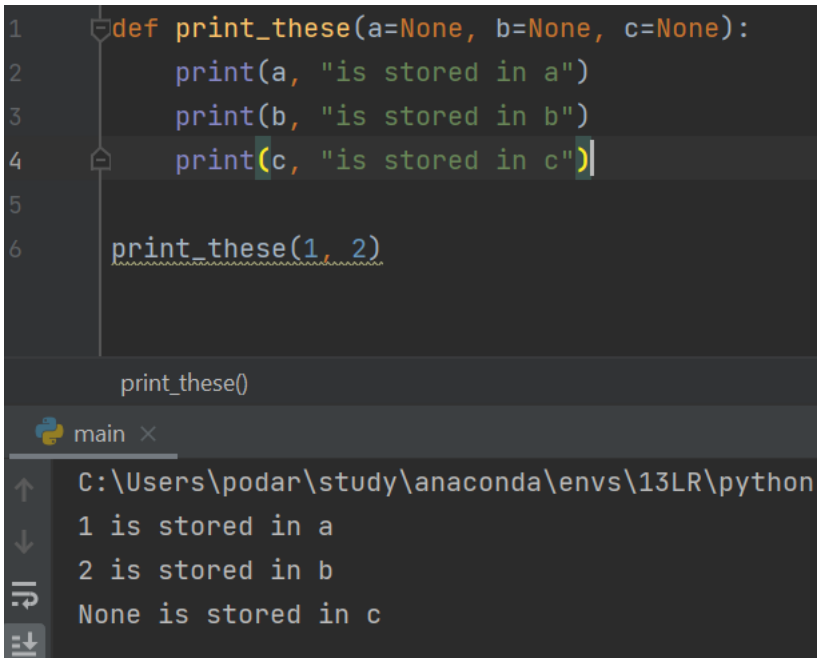

Рисунок 4 – Установление трех опциональных параметров

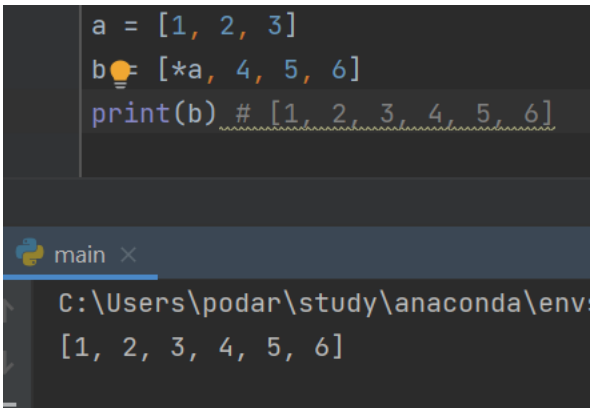

Рисунок 5 – Применение оператора «звездочка»

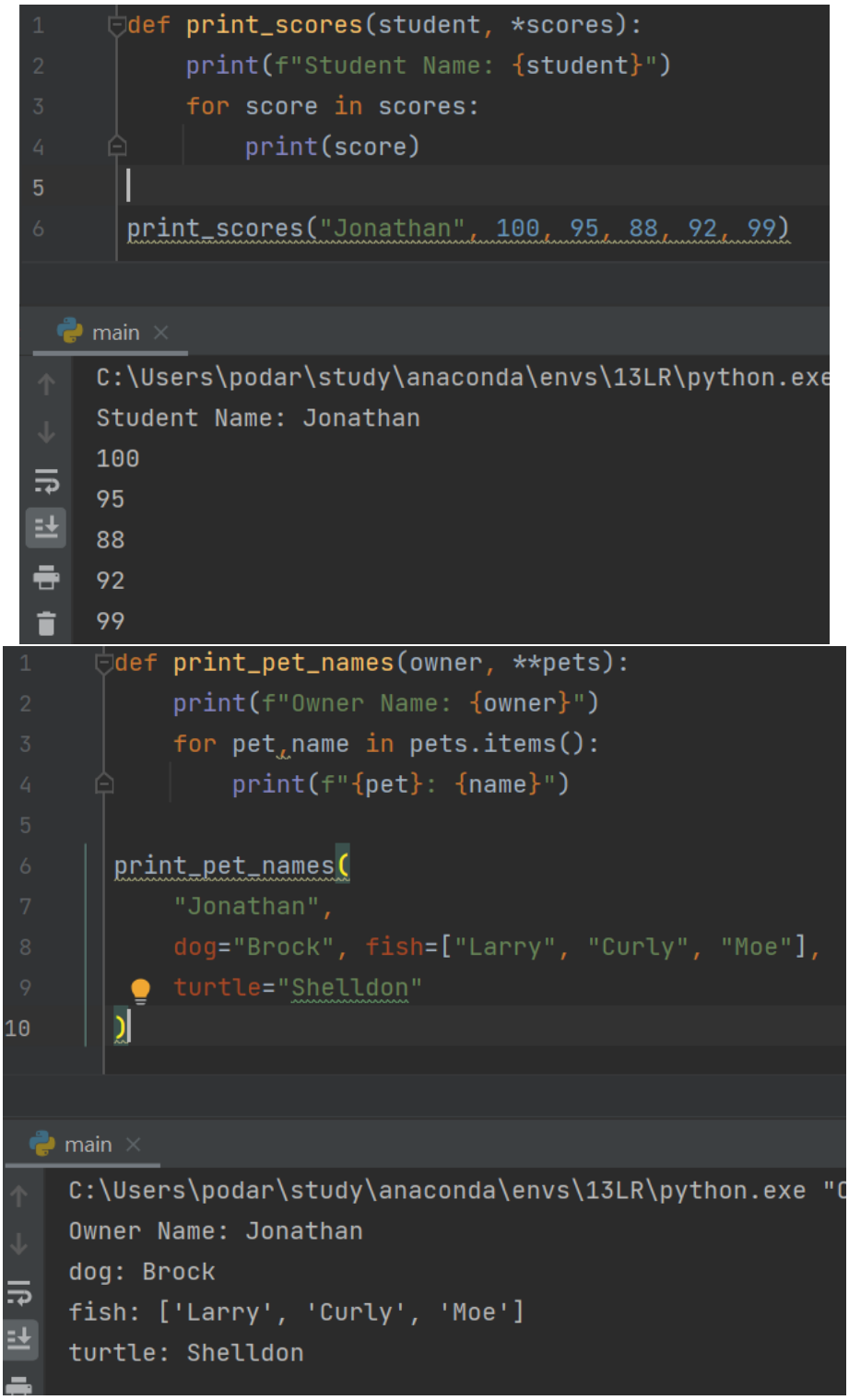

Рисунок 6 – Использование функций *args и *kwargs

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

def median(*args):
    if args:
        values = [float(arg) for arg in args]
        values.sort()

        n = len(values)
        idx = n // 2
        if n % 2:
            return values[idx]
        else:
```

if __name__ == "__main__"

main

```
C:\Users\podar\study\anaconda\envs\13LR\python.exe
None
6.0
4.5
```

Рисунок 8 – Пример

```python
# !/usr/bin/env python3
# -*- coding: utf-8 -*-

def sr_geom(*args):
    if args:
        summa = 0
        for arg in args:
            summa = summa + arg
        n = len(args)
        return summa / n
    else:
        return None

if __name__ == "__main__":
    arguments = list(map(int, input('Введите аргументы: ').split(
    print('Среднее геометрическое значение аргументов: ', sr_geom
```

if __name__ == "__main__"

main

```
C:\Users\podar\study\anaconda\envs\13LR\python.exe "C:/Users/podar/s
Введите аргументы: 5 12 17 27 5 4 19
Среднее геометрическое значение аргументов:  12.714285714285714
```

Рисунок 9 – Решение 1 задания

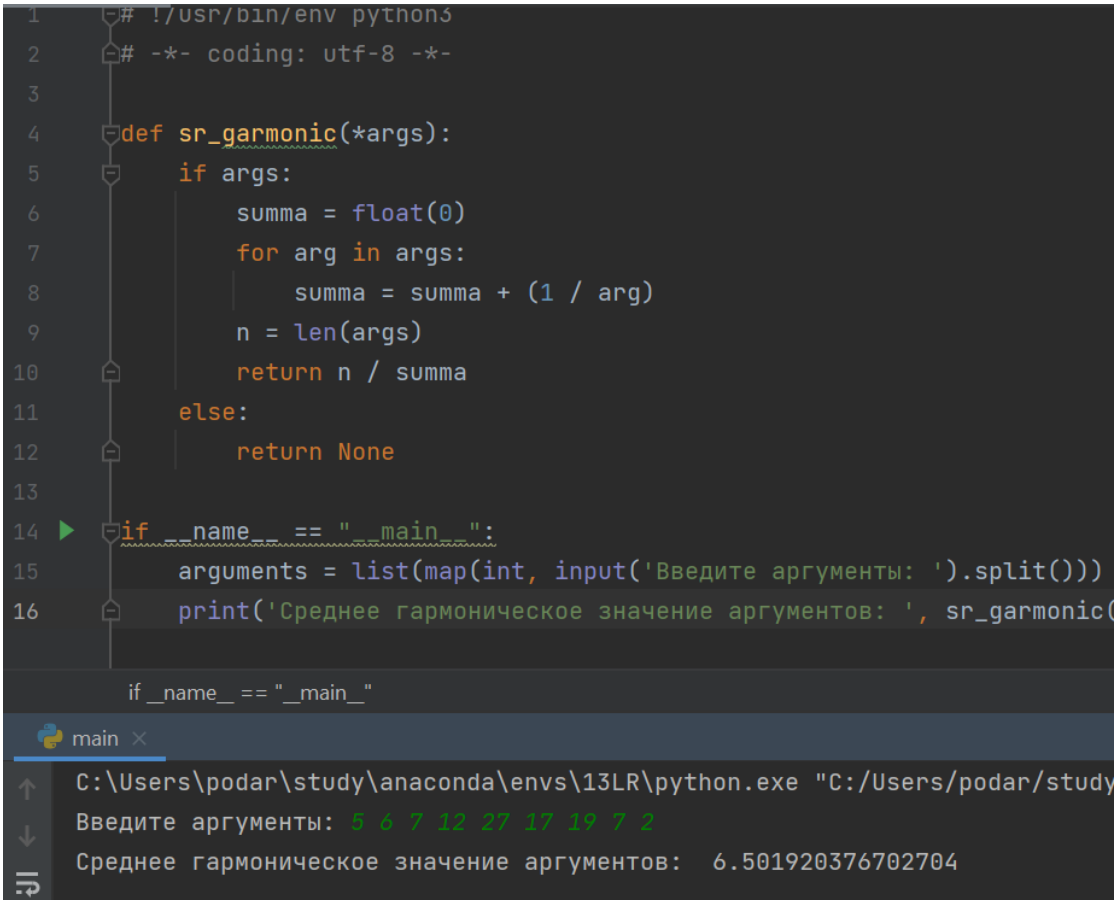

Рисунок 10 – Решение 2 задания

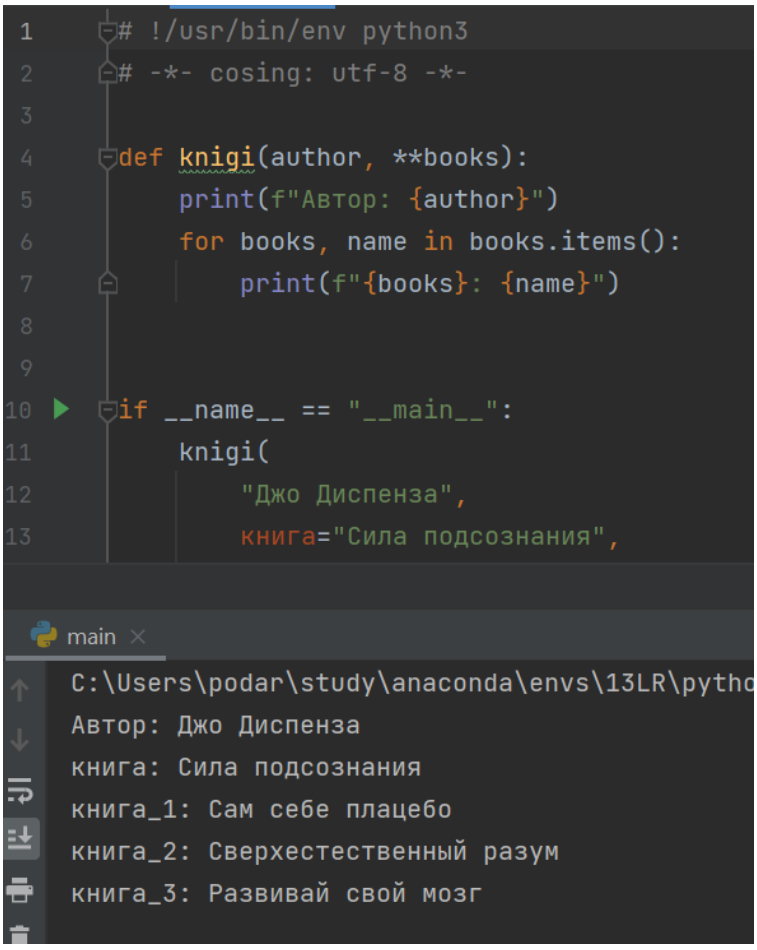

Рисунок 11 – Решение 3 задания

Индивидуальное задание. Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None . Номер варианта определяется по согласованию с преподавателем.
В процессе решения не использовать преобразования конструкции *args в список или иную структуру данных.
Сумму целых частей аргументов, расположенных после последнего отрицательного аргумента.

```
# !/usr/bin/env python3
# -*- cosing: utf-8 -*-


def summ(*args):
    if args:
        summa = 0
        args = reversed(args)
        for argument in args:
            if argument > 0:
                summa += argument
            else:
                break
        return summa
    else:
        return None


if __name__ == "__main__":
    arguments = list(map(float, input('Введите числа: ').split()))
            print('Сумма чисел после последнего отрицательного: ',
summ(*arguments))
```

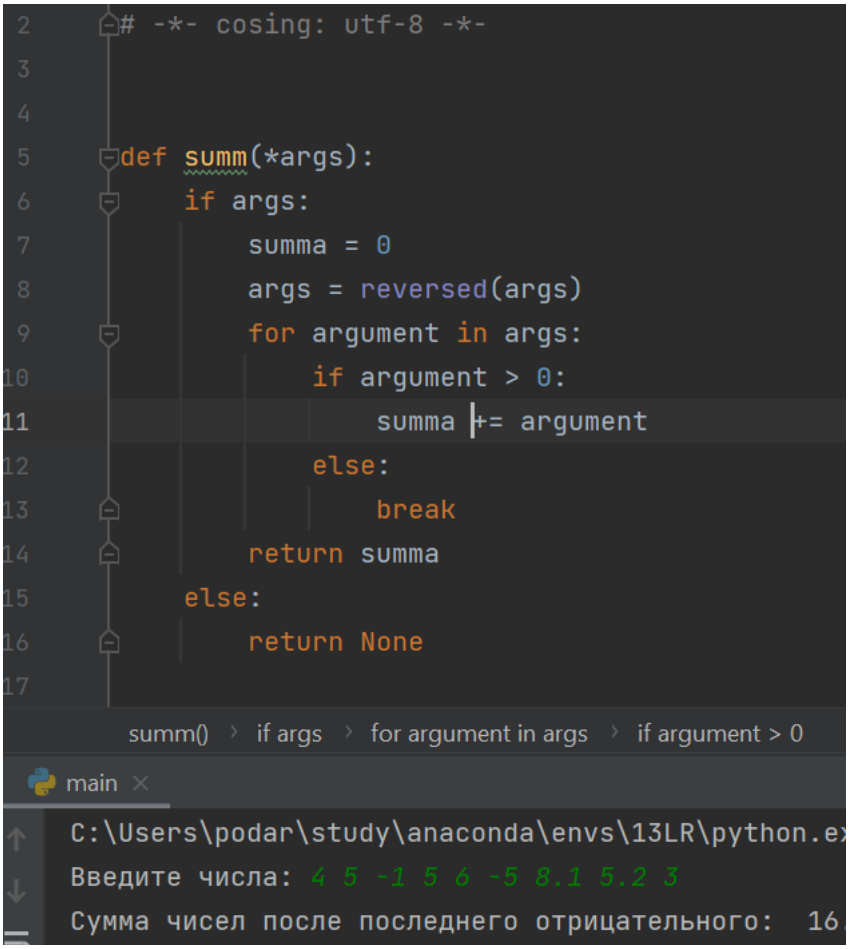

Рисунок 12 – Результат решения идз

Вопросы

1. **Какие аргументы называются позиционными в Python?**
   Это аргументы, передаваемые в вызов в определённой последовательности (на определённых позициях), без указания их имён. Элементы объектов, поддерживающих итерирование, могут использоваться в качестве позиционных аргументов, если их распаковать при помощи *.
2. **Какие аргументы называются именованными в Python?**
   Это аргументы, передаваемые в вызов при помощи имени (идентификатора), либо словаря с его распаковкой при помощи **.
3. **Для чего используется оператор *?**
   Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.
4. **Каково назначение конструкций *args и **kwargs?**
   *args используется для передачи произвольного числа неименованных аргументов функции.
   **kwargs позволяет передавать произвольное число именованных аргументов в функцию.