

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт цифрового развития**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №2.10**

Функции с переменным числом параметров в Python

Выполнил студент группы

ИВТ-б-о-21-1 (2)

Стригалов Д.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

\_\_\_\_\_  
(подпись)

Ставрополь 2022

### Ход работы:

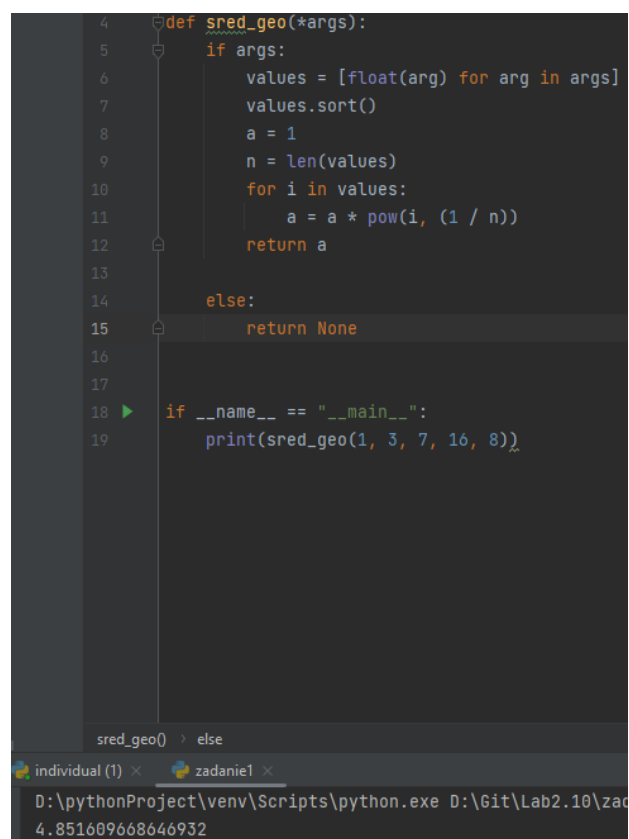
**Цель работы:** приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

### Ход работы:

**Задание №1.** Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов  $a_1, a_2 \dots a_n$ . Если функции передается пустой список аргументов, то она должна возвращать значение None.

$$G = \sqrt[n]{\prod_{k=1}^n a_k}.$$

Рисунок 1 – Формула для задания №1



```
4 def sred_geo(*args):
5     if args:
6         values = [float(arg) for arg in args]
7         values.sort()
8         a = 1
9         n = len(values)
10        for i in values:
11            a = a * pow(i, (1 / n))
12        return a
13
14    else:
15        return None
16
17
18 if __name__ == "__main__":
19     print(sred_geo(1, 3, 7, 16, 8))
```

sred\_geo() > else

individual (1) x zadanie1 x

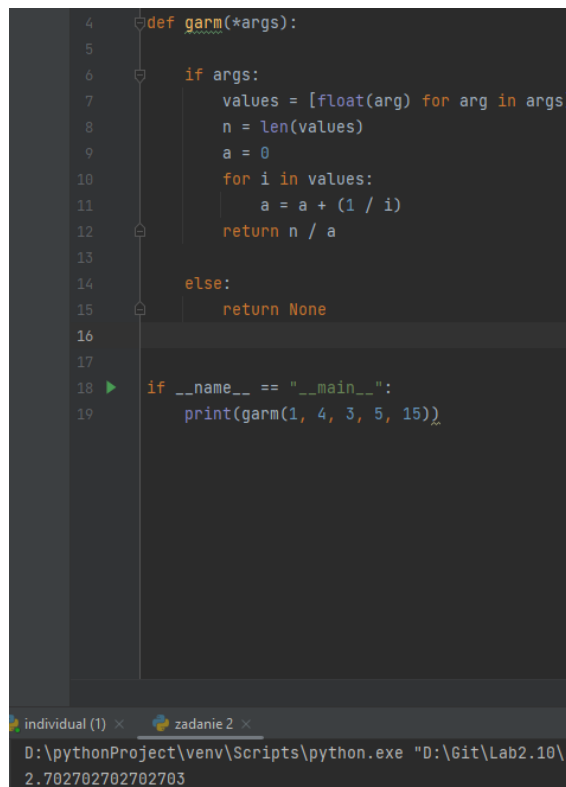
D:\pythonProject\venv\Scripts\python.exe D:\Git\Lab2.10\zad...  
4.851609668646932

Рисунок 2 – Результат работы программы

**Задание №2.** Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов  $a_1, a_2 \dots a_n$ . Если функции передается пустой список аргументов, то она должна возвращать значение None.

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}.$$

Рисунок 3 – Формула для задания №2



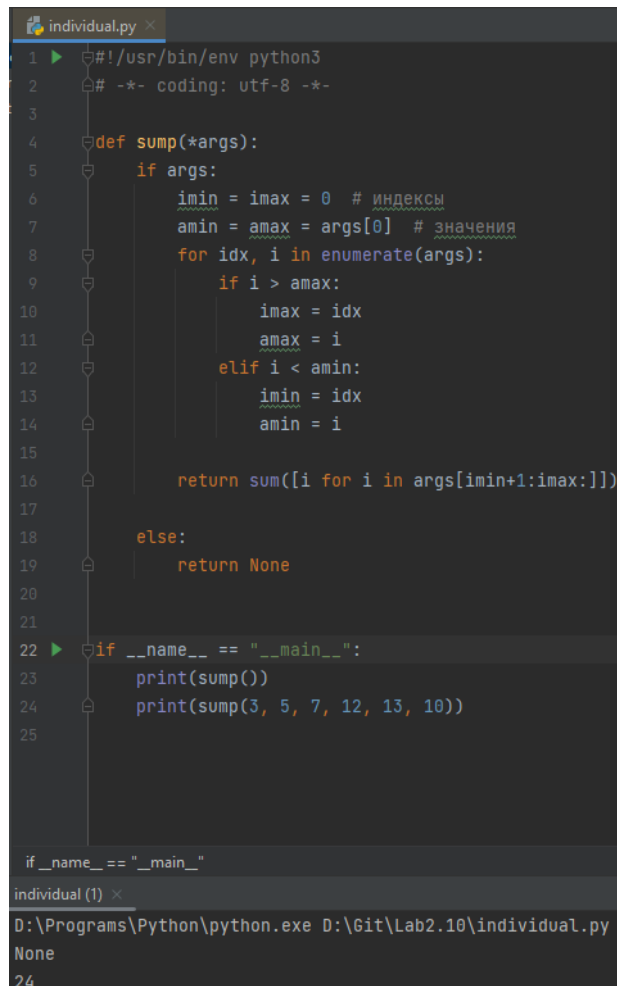
```
4 def garm(*args):
5
6     if args:
7         values = [float(arg) for arg in args]
8         n = len(values)
9         a = 0
10        for i in values:
11            a = a + (1 / i)
12        return n / a
13
14    else:
15        return None
16
17
18 if __name__ == "__main__":
19     print(garm(1, 4, 3, 5, 15))
```

individual (1) × zadanie 2 ×  
D:\pythonProject\venv\Scripts\python.exe "D:\Git\Lab2.10\zadanie2.py"  
2.702702702702703

Рисунок 4 – Результат работы программы

**Индивидуальное задание.** Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение None. В процессе решения не использовать преобразования конструкции `*args` в список или иную структуру данных.

**Задание: 1.** Произведение аргументов, расположенных между максимальным и минимальным аргументами.



```
individual.py x
1  ▶ #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  def sump(*args):
5      if args:
6          imin = imax = 0 # индексы
7          amin = amax = args[0] # значения
8          for idx, i in enumerate(args):
9              if i > amax:
10                 imax = idx
11                 amax = i
12             elif i < amin:
13                 imin = idx
14                 amin = i
15
16             return sum([i for i in args[imin+1:imax:]])
17
18         else:
19             return None
20
21
22  ▶ if __name__ == "__main__":
23      print(sump())
24      print(sump(3, 5, 7, 12, 13, 10))
25
if __name__ == "__main__"
individual (1) x
D:\Programs\Python\python.exe D:\Git\Lab2.10\individual.py
None
24
```

Рисунок 5 – Результат выполнения программы

**Вывод:** в результате выполнения лабораторной работы были приобретены практические навыки и теоретические сведения по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

### Ответы на контрольные вопросы:

#### 1. Какие аргументы называются позиционными в Python?

Аргументы, которые передаются без указания имен называются позиционными, потому что именно по позиции, расположению аргумента, функция понимает, какому параметру он соответствует.

#### 2. Какие аргументы называются именованными в Python?

Аргументы, передаваемые с именами, называются именованными. При вызове функции можно использовать имена параметров из ее определения.

### **3. Для чего используется оператор \*?**

Оператор `*` чаще всего ассоциируется у людей с операцией умножения, но в Python он имеет и другой смысл.

Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

Вот пример:

```
a = [1, 2, 3] b = [*a, 4,  
5, 6] print(b) # [1, 2, 3, 4, 5,  
6]
```

Тут берётся содержимое списка `a`, распаковывается, и помещается в список `b`.

### **4. Каково назначение конструкций \*args и \*\*kwargs?**

Итак, мы знаем о том, что оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы. Знаем мы и о том, что существует два вида параметров функций. А именно, `*args` — это сокращение от «arguments» (аргументы), а `**kwargs` — сокращение от «keyword arguments» (именованные аргументы).

Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины.

**Важно помнить, что «args» — это всего лишь набор символов, которым принято обозначать аргументы. Самое главное тут — это оператор `*`. А то, что именно идёт после него, особой роли не играет. Благодаря использованию `*` мы создали список позиционных аргументов на основе того, что было передано функции при вызове.**

После того, как мы разобрались с `*args`, с пониманием `**kwargs` проблем быть уже не должно.

Имя, опять же, значения не имеет. Главное — это два символа `**`. Благодаря им создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.