

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития
Кафедра инфокоммуникаций

«Функции с переменным числом параметров в Python»

ОТЧЕТ
по лабораторной работе №13
дисциплины
«Основы программной инженерии»

Выполнил:

Мизин Глеб Егорович

2 курс, группа ПИЖ-б-о-21-1,

09.03.04 «Программная

инженерия», направленность

(профиль) «Разработка и

сопровождение программного

обеспечения», очная форма

обучения

(подпись)

Проверил:

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2022 г.

Проработка примера:

```
1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def median(*args):
5     if args:
6         values = [float(arg) for arg in args]
7         values.sort()
8         n = len(values)
9         idx = n // 2
10        if n % 2:
11            return values[idx]
12        else:
13            return (values[idx - 1] + values[idx]) / 2
14    else:
15        return None
16
17
18 ▶ if __name__ == "__main__":
19     print(median())
20     print(median(3, 7, 1, 6, 9))
21     print(median(1, 5, 8, 4, 3, 9))
```

median() > if args

Run: 1_ex ×

F:\GitLab\Lab-2.11\venv\Scripts\python.exe F:\GitLab\Lab-2.10\PyCharm\1_ex.py

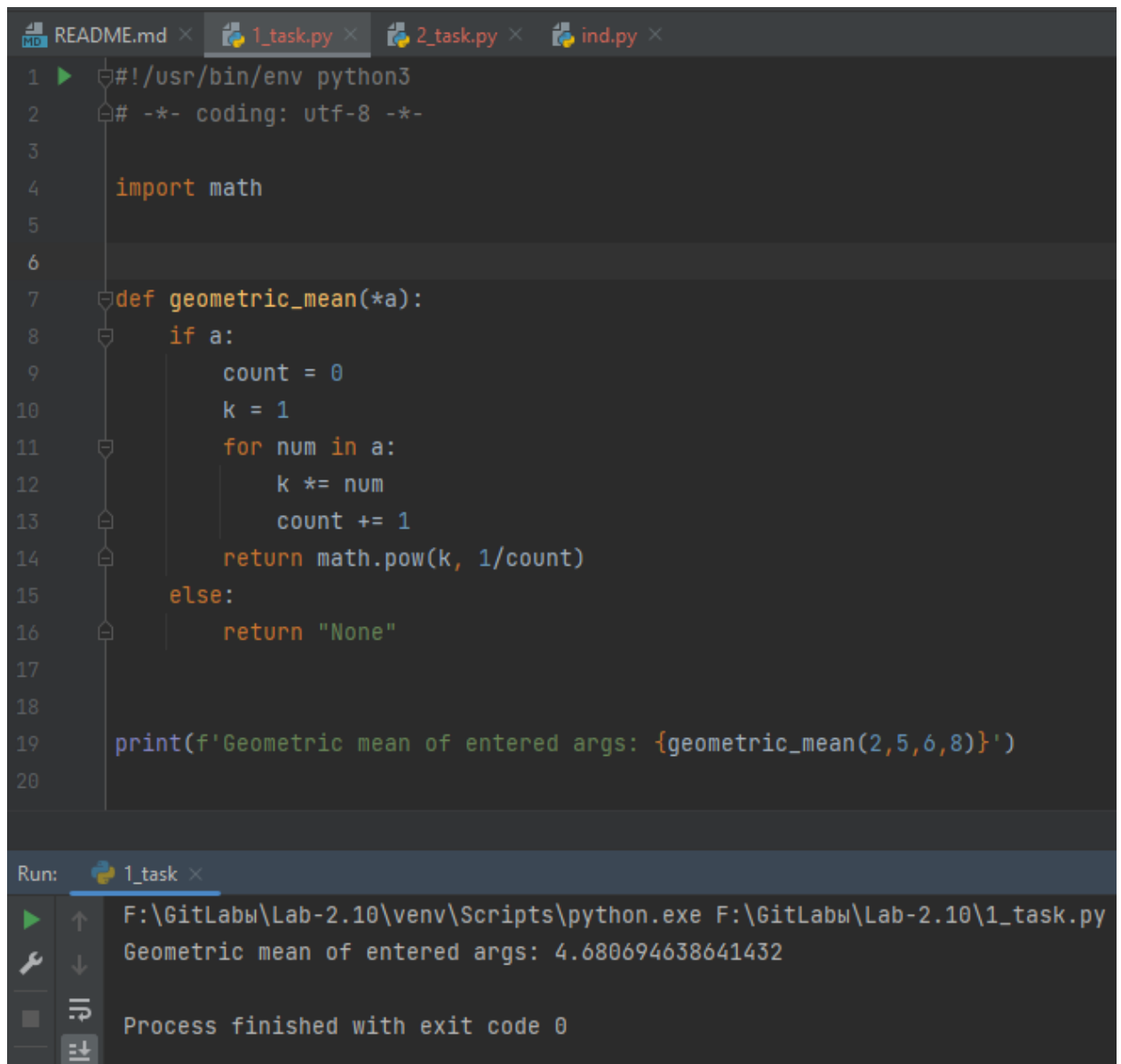
None
6.0
4.5

Рисунок 1 – Код и результат работы примера

Задание №1: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

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

Если функции передается пустой список аргументов, то она должна возвращать значение None .



The image shows a code editor with four tabs: README.md, 1_task.py, 2_task.py, and ind.py. The 1_task.py tab is active, displaying the following Python code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 import math
5
6
7 def geometric_mean(*a):
8     if a:
9         count = 0
10        k = 1
11        for num in a:
12            k *= num
13            count += 1
14        return math.pow(k, 1/count)
15    else:
16        return "None"
17
18
19 print(f'Geometric mean of entered args: {geometric_mean(2,5,6,8)}')
```

Below the code editor, the 'Run' panel shows the execution of 1_task.py. The output is:

```
F:\GitLab\Lab-2.10\venv\Scripts\python.exe F:\GitLab\Lab-2.10\1_task.py
Geometric mean of entered args: 4.680694638641432
Process finished with exit code 0
```

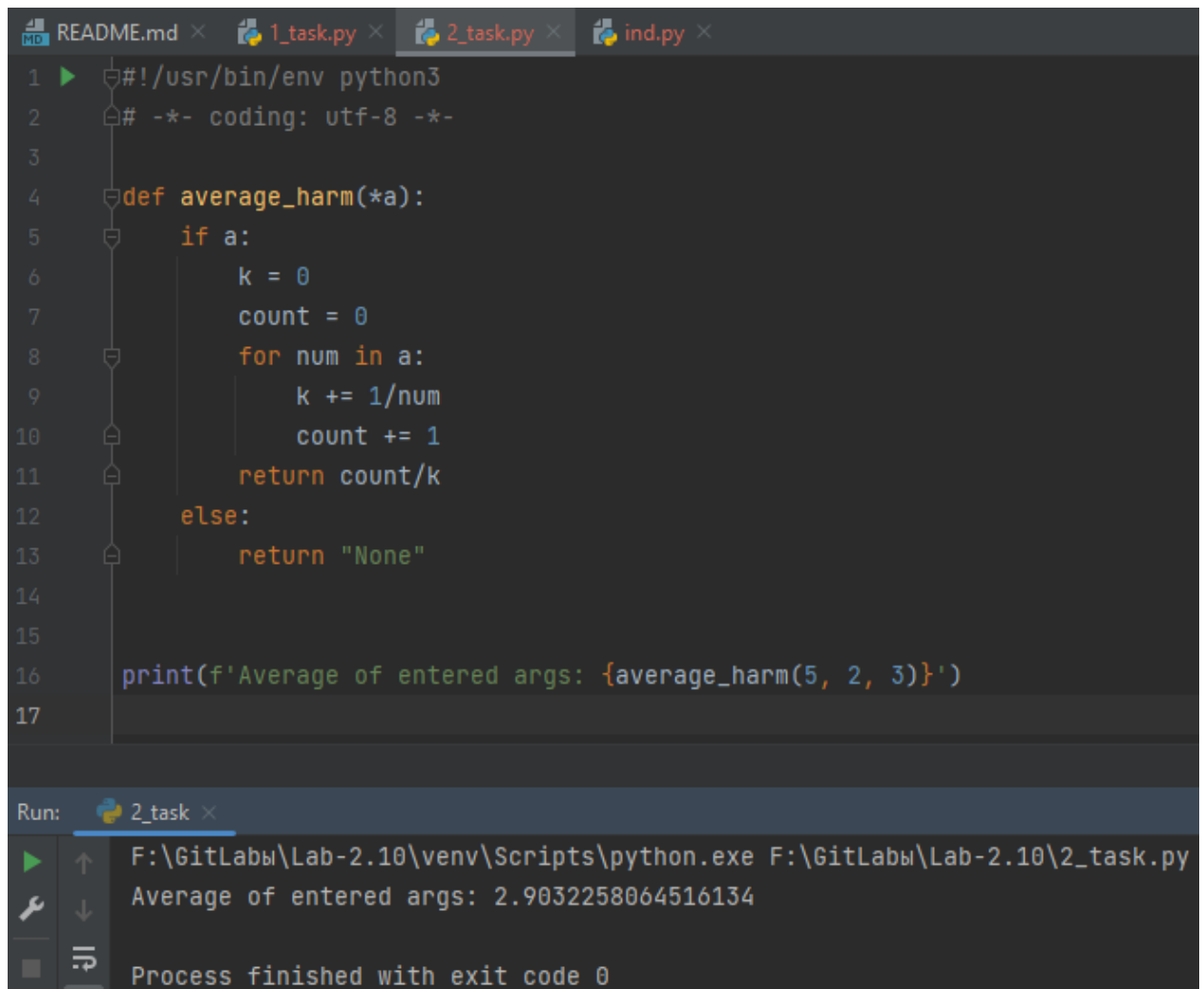
Рисунок 2 – Код и результат работы программы задания №1

Задание №2: решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов

a_1, a_2, \dots, a_n

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

Если функции передается пустой список аргументов, то она должна возвращать значение None .



The image shows a screenshot of a code editor with four tabs: README.md, 1_task.py, 2_task.py (active), and ind.py. The active tab contains the following Python code:

```
1 #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def average_harm(*a):
5     if a:
6         k = 0
7         count = 0
8         for num in a:
9             k += 1/num
10            count += 1
11        return count/k
12    else:
13        return "None"
14
15
16 print(f'Average of entered args: {average_harm(5, 2, 3)}')
```

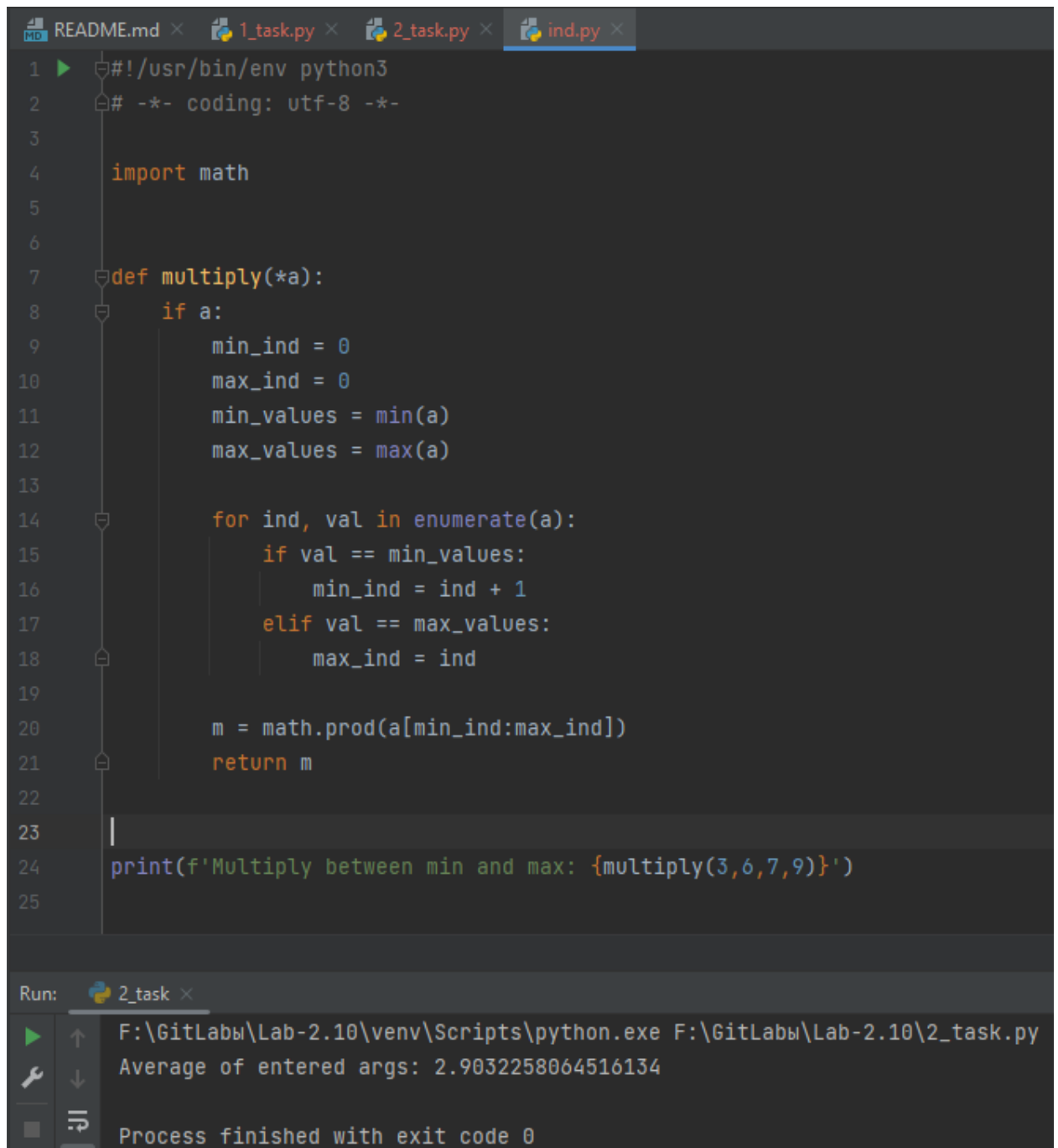
Below the code editor, the 'Run' output is displayed for the file 2_task.py. The output shows the command executed, the resulting string, and the exit code.

```
Run: 2_task x
F:\GitLab\Lab-2.10\venv\Scripts\python.exe F:\GitLab\Lab-2.10\2_task.py
Average of entered args: 2.9032258064516134
Process finished with exit code 0
```

Рисунок 3 – Код и результат работы программы задания №2

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.

Произведение аргументов, расположенных между максимальным и минимальным аргументами



```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import math
5
6
7  def multiply(*a):
8      if a:
9          min_ind = 0
10         max_ind = 0
11         min_values = min(a)
12         max_values = max(a)
13
14         for ind, val in enumerate(a):
15             if val == min_values:
16                 min_ind = ind + 1
17             elif val == max_values:
18                 max_ind = ind
19
20         m = math.prod(a[min_ind:max_ind])
21         return m
22
23
24 print(f'Multiply between min and max: {multiply(3,6,7,9)}')
```

Run: 2_task ×

↑ F:\GitLab\Lab-2.10\venv\Scripts\python.exe F:\GitLab\Lab-2.10\2_task.py
↓ Average of entered args: 2.9032258064516134
⚙
⏏ Process finished with exit code 0

Рисунок 4 – Код и результат работы программы индивидуально задания

Контрольные вопросы

1. Какие аргументы называются позиционными в Python?

Позиционные это аргументы вызванные при помощи оператора *

2. Какие аргументы называются именованными в Python?

Именованные это аргументы вызванные при помощи оператора **

3. Для чего используется оператор * ?

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 ?

```
def printScores(student, *scores):
    print(f"Student Name: {student}")
    for score in scores:
        print(score)
printScores("Jonathan",100, 95, 88, 92, 99)
"""
Student Name: Jonathan
100
95
88
92
99
"""
```

```
def printPetNames(owner, **pets):
    print(f"Owner Name: {owner}")
    for pet,name in pets.items():
        print(f"{pet}: {name}")
printPetNames("Jonathan", dog="Brock", fish=["Larry", "Curly", "Moe"], turtle="Sheldon")
"""
Owner Name: Jonathan
dog: Brock
fish: ['Larry', 'Curly', 'Moe']
turtle: Sheldon
"""
```