

МИНЕСТЕРСТВО НАУКИ ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО - КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт цифрового развития
Кафедра инфокоммуникаций

ОТЧЁТ
по лабораторной работе №5
Дисциплина: «Языки программирования»

Выполнил: студентка 2 курса
группы ИТС-б-о-20-1
Игнатова Елизавета Сергеевна

Проверил:
к.ф.-м.н., доцент
кафедры инфокоммуникаций
Воронкин Роман Александрович

Работа защищена с оценкой: _____

Ставрополь, 2022

Функции с переменным числом параметров в Python

Цель работы: приобрести навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python.

Порядок выполнения

Ссылка на репозиторий: <https://github.com/Nebula139/Sky2.5>

Пример 1

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def print_these(a, b, c):
5          print(a, "is stored in a")
6          print(b, "is stored in b")
7          print(c, "is stored in c")
8
9
10 ▶  if __name__ == '__main__':
11      print_these(1, 2, 3)
```

Рисунок 1 – код примера

```
C:\Users\Лиза\PycharmProjects\pythonProject5.1
1 is stored in a
2 is stored in b
3 is stored in c
```

Рисунок 2 – результат работы кода

Пример 2. Для вызова функции необходимы все три аргумента. Если пропустить хотя бы один из них—будет выдано сообщение об ошибке:

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def print_these(a, b, c):
5          print(a, "is stored in a")
6          print(b, "is stored in b")
7          print(c, "is stored in c")
8
9
10 ▶  if __name__ == '__main__':
11      print_these(1, 2)
12

```

Рисунок 3 – код примера

```

C:\Users\Лиза\PycharmProjects\pythonProject5.1\venv\Scripts\python.exe C:/Users/Лиза/PycharmProjects/pytho
Traceback (most recent call last):
  File "C:\Users\Лиза\PycharmProjects\pythonProject5.1\prim2.py", line 11, in <module>
    print_these(1, 2)
TypeError: print_these() missing 1 required positional argument: 'c'

```

Рисунок 4 – результат работы кода

Пример 3.

```

C:\Users\Лиза\PycharmProjects\pythonPr
1 is stored in a
2 is stored in b
None is stored in c

```

Рисунок 5 – результат работы кода

Пример 4.

```

C:\Users\Лиза\PycharmProjec
1 is stored in a
None is stored in b
3 is stored in c

```

Рисунок 6 – результат работы кода

Пример 5.

```
C:\Users\Лиза\PycharmProjects\pyt  
[1, 2, 3, 4, 5, 6]
```

Рисунок 7– результат работы кода

Пример 6.

```
C:\Users\Лиза\PycharmProjects\pyt  
Student Name: Jonathan  
100  
95  
88  
92  
99
```

Рисунок 8 – результат работы кода

Пример 7.

```
prim7 x  
C:\Users\Лиза\PycharmProjects\pythonProject5.1\  
Owner Name: Jonathan  
dog: Brock  
fish: ['Larry', 'Curly', 'Moe']  
turtle: Shelldon
```

Рисунок 9 – результат работы кода

Пример 8.

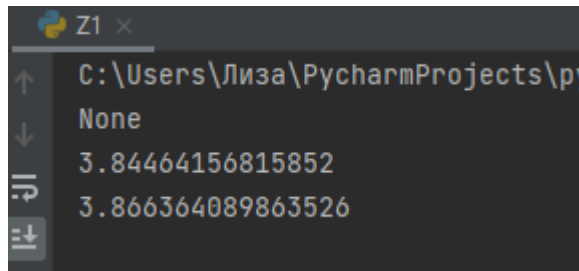
```
prim8 x  
C:\Users\Лиза\PycharmProjects\pythonProj  
None  
6.0  
4.5
```

Рисунок 10 – результат работы кода

8. Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов a_1, a_2, \dots, a_n

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

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



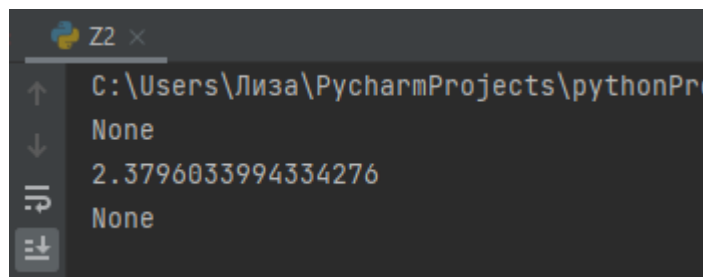
```
Z1 ×
C:\Users\Лиза\PycharmProjects\py
None
3.84464156815852
3.866364089863526
```

Рисунок 11 – результат работы кода

9. Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов a_1, a_2, \dots, a_n

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}. \quad (2)$$

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



```
Z2 ×
C:\Users\Лиза\PycharmProjects\pythonPro
None
2.3796033994334276
None
```

Рисунок 12 – результат работы кода

Индивидуальное задание

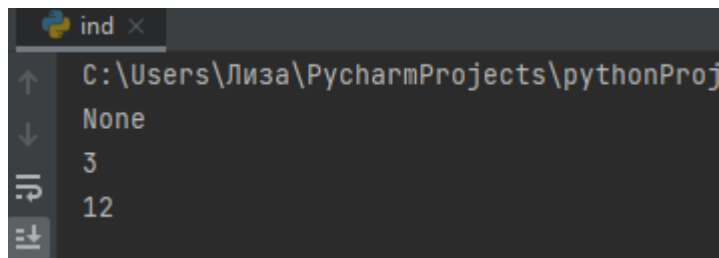
Напишите функцию, принимающую произвольное количество аргументов, и возвращающую требуемое значение. Если функции передается пустой список аргументов, то она должна возвращать значение `None`. Сумму положительных аргументов, расположенных до максимального аргумента.

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def before_max(*args):
5          if args:
6              ans = 0
7              idx = args.index(max(args))
8              for item in args:
9                  if args.index(item) < idx:
10                     ans += item
11             return ans
12         else:
13             return None
14
15
16  ▶  if __name__ == "__main__":
17      print(before_max())
18      print(before_max(2, 1, 0, 8, 7))
19      print(before_max(2, 4, 5, 1, 6, 3))
20

```

Рисунок 13 – код программы



```

ind x
C:\Users\Лиза\PycharmProjects\pythonProj
None
3
12

```

Рисунок 14 – результат работы кода

Контрольные вопросы:

1. Какие аргументы называются позиционными?

Это аргументы, которые при вызове функции надо передать в правильном порядке.

2. Какие аргументы называются именованными?

Если порядок передачи аргументов по каким-то причинам не известен, то можно использовать именованные аргументы. Именованный аргумент представляет собой пару «имя-значение»

3. Для чего используется оператор *?

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

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

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

Вывод: в ходе лабораторной работы были изучены именованные и позиционные аргументы и приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python.