

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

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

Мальцев Н.А. « » _____ 20__ г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил доцент

Кафедры инфокоммуникаций, старший
преподаватель

Воронкин Р.А.

(подпись)

Ставрополь 2022

Рекурсия в языке Python.

Цель работы: приобретение навыков по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.10.

Порядок выполнения работы:

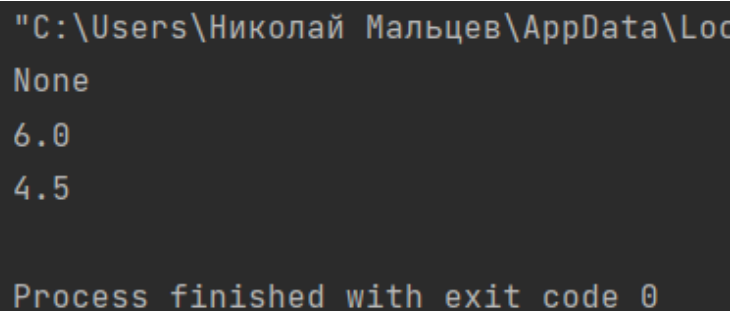
1. Проработка примеров:

Код программы:

```
#!/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:
            return (values[idx - 1] + values[idx]) / 2
    else:
        return None

if __name__ == "__main__":
    print(median())
    print(median(3, 7, 1, 6, 9))
    print(median(1, 5, 8, 4, 3, 9))
```

Результат работы программы:



```
"C:\Users\Николай Мальцев\AppData\Loc
None
6.0
4.5

Process finished with exit code 0
```

Рисунок 1. Результат работы программы из примера 1

2. Выполнение задачи:

Код программы:

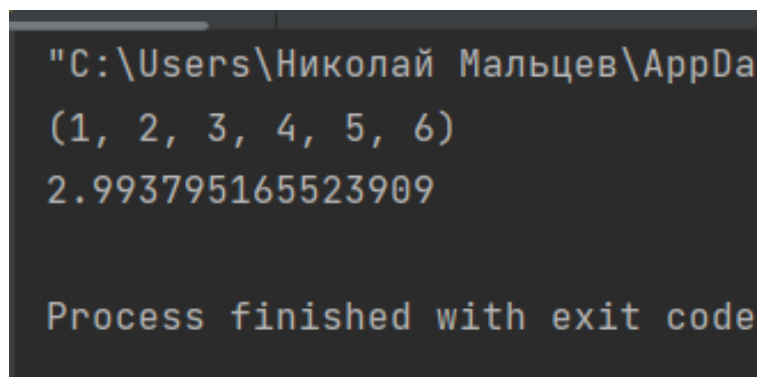
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
```

```
import math

def geo_mean(*values):
    if values:
        temp = 1
        for i in range(0, len(values)):
            temp = temp * values[i]
        return math.pow(temp, 1/len(values))
    else:
        return None

if __name__ == "__main__":
    print(geo_mean(1, 2, 3, 4, 5, 6))
```

Результат работы программы:



```
"C:\Users\Николай Мальцев\AppData
(1, 2, 3, 4, 5, 6)
2.993795165523909

Process finished with exit code
```

Рисунок 2. Результат работы программы к заданию 1

Листинг программы:

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

def mean_gar(*values):
    print(values)
    if values:
        temp = 0
        for i in range(0, len(values)):
            temp = temp + 1/values[i]
        return len(values)/temp
    else:
        return None

if __name__ == "__main__":
    print(mean_gar(1, 2, 3, 4, 5, 6))
```

Результат работы программы:

```
"C:\Users\Николай Мальцев\AppData\Local\Temp\1\python.exe"
(1, 2, 3, 4, 5, 6)
2.4489795918367347

Process finished with exit code 0
```

Рисунок 3. Результат работы программы к заданию 2

Вывод: с использованием декоратора `iru_cache` рекурсивная функция выполняется примерно в 3 раза быстрее, а итеративная в 2.

4. Выполнение индивидуального задания (вариант 11):

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 аргументов, расположенных после первого положительного аргумента.

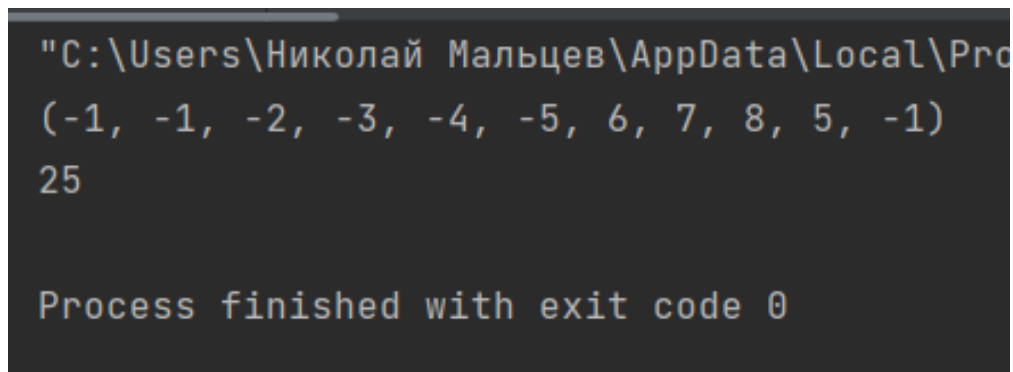
Листинг программы:

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

def sum_positive(*values):
    """
    Сумму аргументов, расположенных
    после первого положительного аргумента.
    """
    if values:
        pos = 0
        temp = 0
        for i in values:
            if i > 0:
                pos = i
        for i in values[pos+1:]:
            temp += i
        return temp
    else:
        return None

if __name__ == '__main__':
    print(sum_positive(-1, -1, -2, -3, -4, -5, 6, 7, 8, 5, -1))
```

Результат выполнения программы:



```
"C:\Users\Николай Мальцев\AppData\Local\Pro  
(-1, -1, -2, -3, -4, -5, 6, 7, 8, 5, -1)  
25  
  
Process finished with exit code 0
```

Рисунок 4. Результат выполнения программы к индивидуальному заданию

Ответы на вопросы:

1. Какие аргументы называются позиционными в Python?

Это аргументы, которые передаются в функцию в соответствии с позицией параметров.

2. Какие аргументы называются именованными в Python?

Именованные аргументы передаются в функцию по их имени, а их позиция не важна.

3. Для чего используется оператор *?

Этот оператор позволяет распаковать объекты, внутри которых хранятся элементы.

4. Каково назначение конструкций *args и *kwargs?

Обе конструкции позволяют передавать в функцию аргументы переменной длины. Аргумент с одной звёздочкой означает именованный аргумент, а с двумя – именованный.

Вывод: в ходе работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.10.