

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

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №13**  
**дисциплины «Программирование на Python»**

Выполнил:  
Середа Кирилл Витальевич  
1 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Роман Александрович

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_

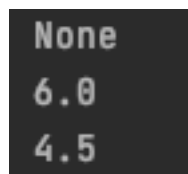
Ставрополь, 2023 г.

Тема: Функции с переменным числом параметров в Python

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

Ход выполнения:

1) Проработал пример - Пример 1 Разработать функцию для определения медианы значений аргументов функции. Если функции передается пустой список аргументов, то она должна возвращать значение None.



```
None
6.0
4.5
```

Рисунок 1 - Результат выполнения программы

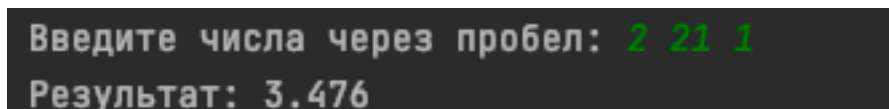
Код программы:

```
#!/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))
```

2) Задача 1 - Решить поставленную задачу: написать функцию, вычисляющую среднее геометрическое своих аргументов



```
Введите числа через пробел: 2 21 1
Результат: 3.476
```

Рисунок 2 - Результат выполнения программы

Код программы:

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

def geometric_mean(*args):
    if args:
        values = list(map(float, args))
        product = 1.0
        for value in values:
            product *= value
        return round(pow(product, 1/len(values)), 4)
    else:
        return None

if __name__ == "__main__":
    input_data = input("Введите числа через пробел: ").split()
    result = geometric_mean(*input_data)
    print("Результат:", result)
```

3) Задача 2 - Решить поставленную задачу: написать функцию, вычисляющую среднее гармоническое своих аргументов

```
Введите числа через пробел: 2 21 1
Результат: 1.9385
```

Рисунок 3 – Результат выполнения программы

Код программы:

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

def calculate_harmonic_mean(*args):
    if args:
        values = list(map(float, args))
        reciprocal_sum = sum(1/value for value in values)
        return round(len(values) / reciprocal_sum, 4)
    else:
        return None

if __name__ == "__main__":
    input_data = input("Введите числа через пробел: ").split()
    result = calculate_harmonic_mean(*input_data)
    print("Результат:", result)
```

4) Выполнил индивидуальное задание – Вариант 1: Произведение аргументов, расположенных между максимальным и минимальным аргументами.

```
Введите числа через пробел: 221 2 2131 2 3 4 1
Результат: 24.0
```

Рисунок 4 – Результат выполнения программы

Код программы:

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

def product_between_min_max(*args):
    if not args:
        return None

    args = [float(arg) for arg in args]

    min_index = args.index(min(args))
    max_index = args.index(max(args))

    start_index, end_index = min(min_index, max_index), max(min_index,
max_index)

    subset = args[start_index + 1:end_index]

    result = 1
    for num in subset:
        result *= num

    return result

if __name__ == "__main__":
    input_data = input("Введите числа через пробел: ").split()

    result = product_between_min_max(*input_data)

    print("Результат:", result)
```

Ответы на вопросы:

1. Позиционные аргументы в Python: В Python аргументы считаются позиционными, когда они передаются функции в том порядке, в котором они определены в самой функции. Возможно также передавать переменное количество позиционных аргументов, используя оператор `*` перед именем аргумента в определении функции.

2. Именованные аргументы в Python: Именованными аргументами в Python называются те, которые передаются функции с указанием имени аргумента, за которым следует значение. Также, как и с позиционными аргументами, можно передавать переменное количество именованных аргументов с использованием оператора `**` перед именем аргумента в определении функции.

3. Оператор `*` в Python: Оператор `*` в Python не только ассоциируется с операцией умножения, но также используется для "распаковки" объектов, содержащих некие элементы. Этот оператор

позволяет извлекать элементы из объектов внутри определенных структур данных.

4. Назначение конструкций `*args` и `kwargs`:

- `*args` (аргументы): Позволяет передавать переменное количество позиционных аргументов в функцию. Объект, переданный с использованием `*args`, может быть распакован, позволяя вызывать функции с различным числом аргументов.

- `**kwargs` (именованные аргументы): Позволяет передавать переменное количество именованных аргументов в функцию. Аналогично `*args`, `**kwargs` позволяет распаковывать словари с аргументами, что облегчает вызов функции с различными именованными параметрами.

Вывод: В результате выполнения лабораторной работы были получены навыки по работе с функциями, поддерживающими переменное количество параметров в Python версии 3.x.