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»

Вариант 7

Выполнил:
Зармухамбетов Булат Эльдарович
2 курс, группа ИВТ-б-о-22-1,
09.03.01 «Информатика и вычислительная техника»,
направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем», очная форма обучения

_____
(подпись)

Руководитель практики:
Воронкин Р. А.

_____
(подпись)

Отчет защищен с оценкой_____ Дата защиты_____

Ставрополь, 2023

Тема: Функции с переменным числом параметров в Python

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

Пример. Разработать функцию для определения медианы значений аргументов функции. Если функции передается пустой список аргументов, то она должна возвращать значение None . Медианой (серединой) набора чисел называется число стоящее посередине упорядоченного по возрастанию ряда чисел. Если количество чисел в ряду чётное, то медианой ряда является полусумма двух стоящих посередине чисел. Применяется в математической статистике — число, характеризующее выборку (например, набор чисел), также используется для вычисления медианной зарплаты.

Листинг:

```
#!/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))
```

```
C:\Users\User\PycharmProjects\Python_laba_13\venv\Scri
None
6.0
4.5

Process finished with exit code 0
```

Рисунок 1. Результат программы

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

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

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

Рисунок 2. Содержание задания №1

Листинг:

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


def geometric_sr(*args):
    """
    Функция ищет среднее геометрическое аргументов
    """
    if args:
        values = [float(arg) for arg in args]
        count = 1
        for value in values:
            count *= value
        return round(pow(count, 1/len(values)), 4)
    else:
        return None


if_name == "_main_":
    print(geometric_sr(4, 8, 16))
    print(geometric_sr(3, 9, 27))
    print(geometric_sr (2, 3, 4))
    print(geometric_sr(31, 12, 32))
```

```
C:\Users\User\PycharmProjects\Python_laba_13\venv\S
8.0
9.0
2.8845
22.8331

Process finished with exit code 0
```

Рисунок 3. Результат программы

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

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

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

Рисунок 4. Содержание задания №2

Листинг:

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


def harmonica(*args):
    """
    Функция находит среднее гармоническое аргументов
    """
    if args:
        values = [float(arg) for arg in args]
        count = 0
        for value in values:
            count += 1/value
        return round(len(values)/count, 4)
    else:
        return None


if_name == "_main_":
    print(harmonica(4, 8, 16))
    print(harmonica(3, 9, 27))
    print(harmonica(2, 3, 4))
    print(harmonica(31, 12, 32))
    print(harmonica(5))
    print(harmonica())
```

```
C:\Users\User\PycharmProjects\Python_laba_13\venv\Sc
6.8571
6.2308
2.7692
20.4302
5.0
None
```

Рисунок 5. Результат программы

Задание 3. Самостоятельно подберите или придумайте задачу с переменным числом именованных аргументов. Приведите решение этой задачи.

Листинг:

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


def print_values(**kwargs):
    """
    Функция выводит на экран аргументы (с их названиями), которые переданы в функцию
    """
    for key, value in kwargs.items():
        print(f"{key}: {value}")
```

```
if __name__ == '__main__':
    print_values(name="Bulat", age=20, city="Stavropool")
```

Рисунок 6. Результат 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. Номер варианта определяется по согласованию с преподавателем. В процессе решения не использовать преобразования конструкции *args в список или иную структуру данных. Сумму аргументов, расположенных между первым и вторым отрицательными аргументами.

Листинг:

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


def get_sum_between_negatives(*args):
    negative1_index = -1
    negative2_index = -1
    total_sum = 0

    for i, arg in enumerate(args):
        if arg < 0:
            if negative1_index == -1:
                negative1_index = i
            else:
                negative2_index = i
                break

    if negative1_index == -1 or negative2_index == -1:
        return None

    for arg in args[negative1_index + 1:negative2_index]:
        total_sum += arg

    return total_sum


if __name__ == '__main__':
    print(get_sum_between_negatives(1, 2, -3, 4, -5, 6))  # Output: 4
    print(get_sum_between_negatives(-1, -2, -3))  # Output: None
    print(get_sum_between_negatives())  # Output: None
```

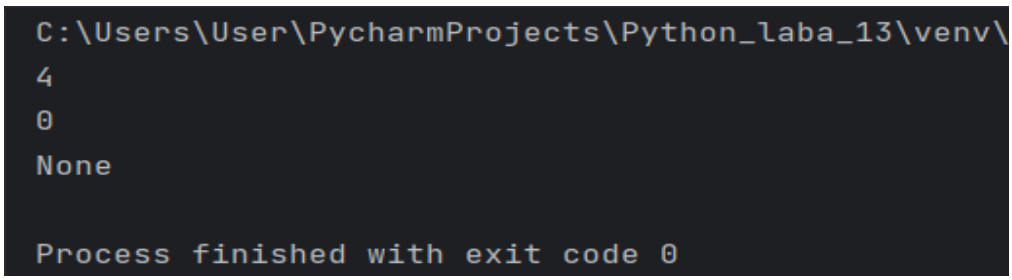

Рисунок 7. Результат программы

Вывод: в ходе выполнения данной лабораторной работы были приобретены навыки по работе с функциями с переменным числом параметров при написании программ с помощью языка программирования Python версии 3.x.

Ответы на контрольные вопросы

1. Позиционные аргументы в Python - это аргументы, которые передаются в функцию в порядке, определенном при объявлении функции. При вызове функции значения аргументов передаются по позиции.

Например, в функции def example(arg1, arg2), arg1 и arg2 являются позиционными аргументами.

2. Именованные аргументы в Python - это аргументы, которые передаются в функцию с указанием имени аргумента (ключа). При вызове функции значения аргументов передаются в виде пар ключ-значение.

Например, в функции def example(arg1, arg2), arg1 и arg2 являются позиционными аргументами, но при вызове функции они могут быть переданы в виде именованных (keyword) аргументов example(arg1=10, arg2=20).

3. Оператор * используется для распаковки элементов из итерируемого объекта (например, списка или кортежа) в аргументы функции.

Например, если у нас есть список [1, 2, 3] и мы вызываем функцию example(*[1, 2, 3]), то значения из списка будут переданы в функцию в виде отдельных аргументов, т.е. func(1, 2, 3).

4. Конструкции *args и **kwargs в функции используются для того, чтобы позволить функции принимать переменное число аргументов.

*args позволяет функции принимать произвольное число позиционных аргументов, которые передаются в виде кортежа.

**kwargs позволяет функции принимать произвольное число именованных аргументов, которые передаются в виде словаря.

Эти конструкции обычно используются, когда нам неизвестно заранее, сколько аргументов может быть передано функции, или когда мы хотим предоставить пользователю возможность передавать любое количество аргументов.