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 23**

|  |  |  |
| --- | --- | --- |
|  | Выполнил:  Омонкулов Исомиддин Валижон угли  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, 2024

**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))

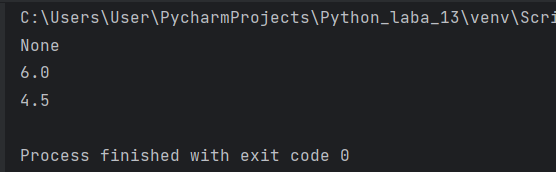


Рисунок 1. Результат программы

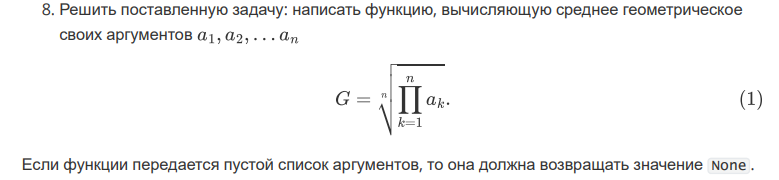


Рисунок 2. Содержание задания №1

**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))

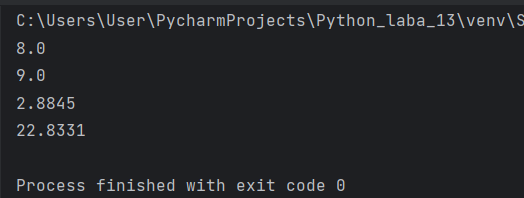


Рисунок 3. Результат программы

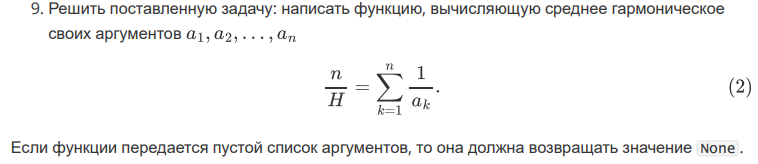


Рисунок 4. Содержание задания №2

**Листинг:**

#!/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())

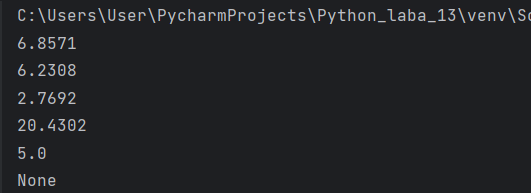


Рисунок 5. Результат программы

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

**Листинг:**

#!/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="Isomiddin", age=19, 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

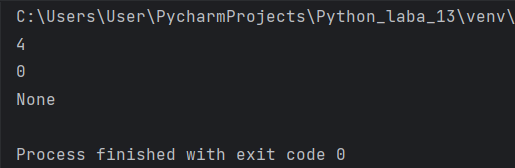


Рисунок 7. Результат программы

**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.