

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

Кафедра инфокоммуникаций

«Основы ветвления Git»

Отчет по лабораторной работе № 2.10

по дисциплине «Основы программной инженерии»

Выполнил студент группы ПИЖ-б-о-21-1

Образцова М. Д.. «17» декабря 2022г.

Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____

(подпись)


Ставрополь 2022

МЕТОДИКА И ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Создала репозиторий в GitHub «OPI_LR_13» в который добавила .gitignore для работы с IDE PyCharm с ЯП Python, выбрала лицензию MIT.

Owner *

Repository name *


 obrMaria ▾

 /


OPI_LR_13 ✓

Great repository names are short and memorable. Need inspiration? How about [congenial-spork?](#)

Description (optional)

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)


Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▾

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).

 You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

```

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_13 (main)
$ git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/M/desktop/OPI_LR_13/.git/hooks]

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_13 (develop)
$ git add

```

Рисунок 2 – Организация репозитория в соответствии с моделью ветвления git-flow

```

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_13 (develop)
$ git add .

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_13 (develop)
$ git commit -m "gitignore"
[develop 063f338] gitignore
1 file changed, 140 insertions(+), 2 deletions(-)

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_13 (develop)
$ git status
On branch develop
nothing to commit, working tree clean

M@DESKTOP-UVM9NOL MINGW64 ~/desktop/OPI_LR_13 (develop)
$

```

Рисунок 3 – Изменение .gitignore

2. Проработаны примеры лабораторной работы.

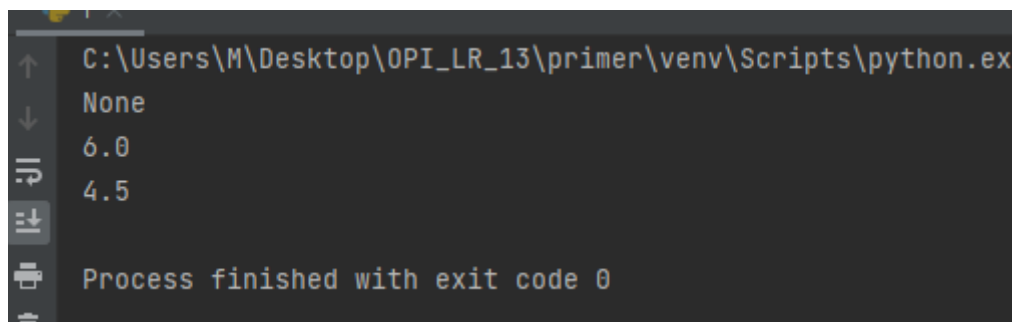
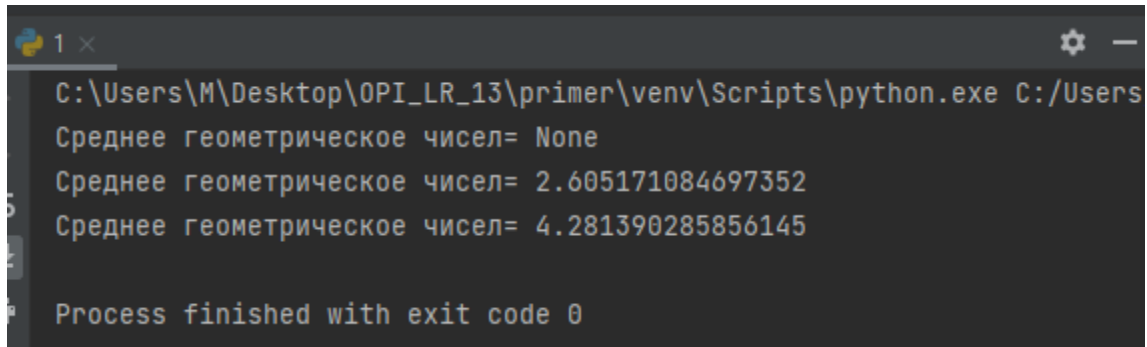


Рисунок 4 – результат выполнения программы

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

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



```
1 x
C:\Users\M\Desktop\OPI_LR_13\primer\venv\Scripts\python.exe C:/Users
Среднее геометрическое чисел= None
Среднее геометрическое чисел= 2.605171084697352
Среднее геометрическое чисел= 4.281390285856145

Process finished with exit code 0
```

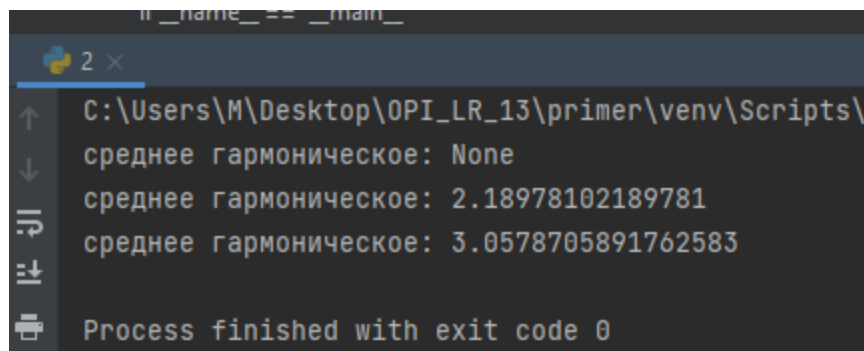
Рисунок 5 – Вывод программы задания

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

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

$$\frac{n}{H} = \sum_{k=1}^n \frac{1}{a_k}.$$

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



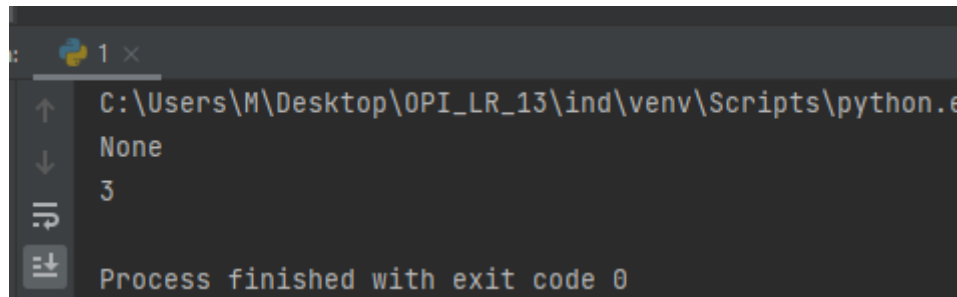
```
if __name__ == '__main__':
2 x
C:\Users\M\Desktop\OPI_LR_13\primer\venv\Scripts\
среднее гармоническое: None
среднее гармоническое: 2.18978102189781
среднее гармоническое: 3.0578705891762583

Process finished with exit code 0
```

Рисунок 5 – Вывод программы задания

5. Решите индивидуальное задание согласно своего варианта.

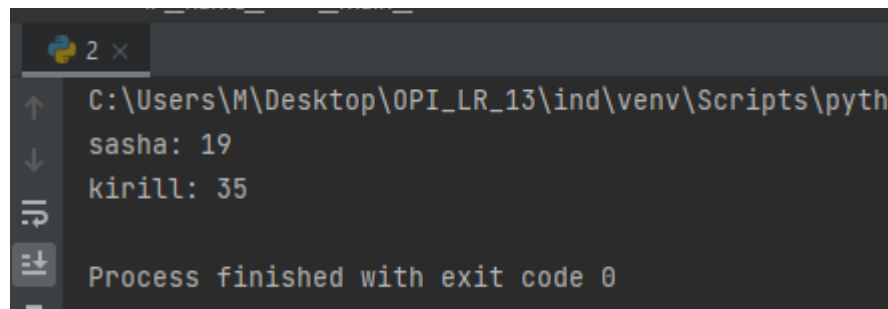
20(3) Сумму аргументов, расположенных между первым и последним нулевыми аргументами.



```
1 x
C:\Users\M\Desktop\OPI_LR_13\ind\venv\Scripts\python.exe
None
3
Process finished with exit code 0
```

Рисунок 5 – Вывод программы задания

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



```
2 x
C:\Users\M\Desktop\OPI_LR_13\ind\venv\Scripts\python.exe
sasha: 19
kirill: 35
Process finished with exit code 0
```

Рисунок 6 – Вывод программы задания

Контр. вопросы и ответы на них:

1. Какие аргументы называются позиционными в Python?

Это аргументы, передаваемые в вызов в определённой последовательности (на определённых позициях), без указания их имён. Элементы объектов, поддерживающих итерирование, могут использоваться в качестве позиционных аргументов, если их распаковать при помощи `*`.

2. Какие аргументы называются именованными в Python?

Это аргументы, передаваемые в вызов при помощи имени (идентификатора), либо словаря с его распаковкой при помощи `**`.

3. Для чего используется оператор `*`?

Функция также может принимать переменное количество позиционных аргументов, тогда перед именем ставится `*`.

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.