

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №2.10 по дисциплине основы  
программной инженерии**

Выполнил:

Кожухов Филипп Денисович,  
2 курс, группа ПИЖ-б-о-20-1,

Проверил:

Доцент кафедры  
прикладной математики и  
компьютерной  
безопасности, Воронкин Р.А.

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

Ставрополь, 2021 г.

## ВЫПОЛНЕНИЕ:

```
1 ▶ #!/usr/bin/env python3
2   # -*- coding: utf-8 -*-
3
4
5   def mid_geom(*args):
6       if args:
7           multiplication = 1
8           values = [float(arg) for arg in args]
9           n = len(values)
10          for elem in values:
11              multiplication *= elem
12          return multiplication ** (1 / n)
13       else:
14           return None
15
16
17 ▶ if __name__ == "__main__":
18     arguments = [float(i) for i in input("Enter the arguments: ").split()]
19     print(f"The geometric mean of these args is: {mid_geom(*arguments)}")
```

Рисунок 1 - Код задачи №1

```
Enter the arguments: 2 5 14 2 6
The geometric mean of these args is: 4.416333444896913
```

Рисунок 2 - Вывод программы

```

1 ▶  #!/usr/bin/env python3
2    # -*- coding: utf-8 -*-
3
4
5    def mid_harm(*args):
6        if args:
7            values = [float(arg) for arg in args]
8            n = len(values)
9            sum_of_reversed = 0
10           for value in values:
11               sum_of_reversed += (1 / value)
12           return n / sum_of_reversed
13        else:
14            return None
15
16
17 ▶  if __name__ == "__main__":
18     arguments = [float(i) for i in input("Enter the arguments: ").split()]
19     print(f"The harmonic mean of these args is: {mid_harm(*arguments)}")

```

Рисунок 3 - Код задачи №2

```

Enter the arguments: 1 4 62 6 3
The harmonic mean of these args is: 2.8310502283105023

Process finished with exit code 0
|

```

Рисунок 4 - Вывод программы

```

1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5  def min_laboratories(**kwargs):
6      """ Минимальное количество сданных работ по ОПИ
7
8      На вход поступает имя студента и количество сданных работ
9
10     Функция выводит студента с наименьшим количеством работ
11
12     """
13     if kwargs:
14         min_labs = min(kwargs.values())
15         for student, labs in kwargs.items():
16             if labs == min_labs:
17                 print(
18                     f"The least laboratories ({labs}) "
19                     f"has the student: {student}"
20                 )
21     else:
22         return None
23
24
25  ▶  if __name__ == "__main__":
26      min_laboratories(Elena=6, Dima=12, Oleg=3, Trofim=7)
27      min_laboratories(Sasha=9, Vitya=15, Aleksey=14)
28      min_laboratories(Efim=13)

```

Рисунок 5 - Код задачи №3

```

The least laboratories (3) has the student: Oleg
The least laboratories (9) has the student: Sasha
The least laboratories (13) has the student: Efim

Process finished with exit code 0

```

Рисунок 6 - Вывод программы

# ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

## Вариант 5

```
1  ▶ #!/usr/bin/env python3
2    # -*- coding: utf-8 -*-
3
4
5    def positive_sum(*args):
6        if args:
7            i = 0
8            for index, arg in enumerate(args):
9                if arg > 0:
10                   i = index
11            pos_s = sum(arg for index, arg in enumerate(args) if index < i)
12            return pos_s
13        else:
14            return None
15
16
17  ▶ if __name__ == "__main__":
18      arguments = [int(i) for i in input("Enter the arguments: ").split()]
19      print("The sum of the arguments before the last positive element is: "
20            f"{positive_sum(*arguments)}")
21  = )
22  |
```

Рисунок 7 - Код индивидуального задания

```
Enter the arguments: 1 52 6 3 7
The sum of the arguments before the last positive element is: 62

Process finished with exit code 0
```

Рисунок 8 - Вывод программы

## ОТВЕТЫ НА ВОПРОСЫ

1. Какие аргументы называются позиционными в Python?

При вызове функций значения в такие аргументы подставляются согласно позиции имён аргументов в определении функции.

2. Какие аргументы называются именованными в Python?

Это аргументы, передаваемые в функцию вместе с именем.

3. Для чего используется оператор `*` ?

Этот оператор позволяет «распаковывать» объекты, внутри которых хранятся некие элементы.

Пример:

```
a = [1, 2, 3]
```

```
b = [*a, 4, 5, 6]
```

```
print(b) # [1, 2, 3, 4, 5, 6]
```

4. Каково назначение конструкций `*args` и `**kwargs` ?

`*args` — это сокращение от «arguments» (аргументы), а `**kwargs` — сокращение от «keyword arguments» (именованные аргументы). Каждая из этих конструкций используется для распаковки аргументов соответствующего типа, позволяя вызывать функции со списком аргументов переменной длины. Благодаря использованию `*` мы создаём список позиционных аргументов на основе того, что было передано функции при вызове. Благодаря символам `**` создаётся словарь, в котором содержатся именованные аргументы, переданные функции при её вызове.