

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 №13 по дисциплине
«Основы программной инженерии»

Выполнил студент
2 курса, группы ПИЖ-б-о-20-1
Тотубалина С.С.

Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2021 г

Ход работы

```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4      def median(*args):
5          if args:
6              values = [float(arg) for arg in args]
7              values.sort()
8
9              n = len(values)
10             idx = n // 2
11             if n % 2:
12                 return values[idx]
13             else:
14                 return (values[idx - 1] + values[idx]) / 2
15
16         else:
17             return None
18
19
20  ▶  if __name__ == "__main__":
21      print(median())
22      print(median(3, 7, 1, 6, 9))
23      print(median(1, 5, 8, 4, 3, 9))
```

Рис. 1 – код программы lab.13_ex.1.py

```
None
6.0
4.5

Process finished with exit code 0
```

Рис. 2 – результат работы программы lab.13_ex.1.py

```

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

```

Рис. 3 – код программы lab.13_ex.2.py

```

Enter the arguments: 5 4 7 6 2 4 7 5
The geometric mean of these arguments is: 4.692774272427004

Process finished with exit code 0

```

Рис. 4 – результат работы программы lab.13_ex.2.py

```

Enter the arguments:
The geometric mean of these arguments is: None

Process finished with exit code 0

```

Рис. 5 – результат работы программы lab.13_ex.2.py при введении пустого множества

```

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 arguments is: {mid_harm(*arguments)}")

```

Рис. 6 – код программы lab.13_ex.3.py

```

Enter the arguments: 2 4 6 9 8 2
The harmonic mean of these arguments is: 3.6302521008403366

Process finished with exit code 0

```

Рис. 7 – результат работы программы lab.13_ex.3.py

```

Enter the arguments:
The harmonic mean of these arguments is: None

Process finished with exit code 0

```

Рис. 8 – результат работы программы lab.13_ex.3.py при введении пустого множества

```

1 ▶ 1 #!/usr/bin/env python3
2   2 # -*- coding: utf-8 -*-
3
4
5   5 def book(author, **books):
6       6     print(f"Author: {author}")
7       7     for books, name in books.items():
8           8         print(f"{name}")
9
10
11 ▶ 11 if __name__ == '__main__':
12     12     book(
13         13         "Э. М. Ремарк",
14         14         book1="На западном фронте без перемен",
15         15         book2="Триумфальная арка",
16         16         book3="Три товарища"
17     17     )
18     18     book(
19         19         "М. Ю. Лермонтов",
20         20         book1="Герой нашего времени"
21     21     )

```

Рис. 9 – код программы lab.13_ex.4.py

```

Author: Э. М. Ремарк
На западном фронте без перемен
Триумфальная арка
Три товарища
Author: М. Ю. Лермонтов
Герой нашего времени

Process finished with exit code 0

```

Рис. 10 – пример работы программы lab.13_ex.4.py

```

1 ▶ #!/usr/bin/env python3
2 # -*- coding: utf-8 -*-
3
4 def after_sum(*args):
5     if args:
6         left = 0
7         r = 0
8         mult = 1
9         flagl = False
10        flagr = False
11        for index, arg in enumerate(args):
12            if arg == 0:
13                if not flagl:
14                    left = index
15                    flagl = True
16                elif not flagr:
17                    r = index
18                    flagr = True
19            if not (flagr * flagl):
20                return None
21            else:
22                for index, arg in enumerate(args):
23                    if (index > left) and (index < r):
24                        mult = mult * arg
25                return mult
26        else:
27            return None
28
29
30 ▶ if __name__ == "__main__":
31     arguments = [int(i) for i in input().split()]
32     arguments.reverse()
33     print(after_sum(*arguments))

```

Рис. 11 – код программы individual_13.py (Вариант № 22)

```

4 0 9 7 0 3 0
3
Process finished with exit code 0

```

Рис. 12 – результат работы программы individual_13.py (Вариант № 22)

```
1 0 2 3  
None  
  
Process finished with exit code 0
```

Рис. 13 – результат работы программы individual_13.py (Вариант № 22) при вводе пустого множества

Ответы на вопросы:

1. Какие аргументы называются позиционными в Python?

Позиционные аргументы - это такие аргументы, значение которых будет зависеть от их позиции. Пример: `def test(a, b) -> a, b` – это позиционные аргументы. Именно по позиции, расположению аргумента, функция понимает, какому параметру он соответствует.

2. Какие аргументы называются именованными в Python?

Аргументы, передаваемые с именами, называются именованными. При вызове функции можно использовать имена параметров из ее определения.

3. Для чего используется оператор `*` ?

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

Оператор «звёздочка» в Python способен «вытаскивать» из объектов составляющие их элементы. Существует два вида параметров функций, а именно: `*args` — это сокращение от «arguments» (аргументы), а `**kwargs` — сокращение от «keyword arguments» (именованные аргументы).