

Ссылка на Replit: <https://replit.com/@melnik3570/LR1?v=1>

Задание 1

Python 3.12.7 64-bit ~\AppData\Local\Programs\Python\Python312\python.exe

Python 3.11.7 64-bit C:\msys64\ucrt64\bin\python3.exe

```
Usage:
  pip <command> [options]

Commands:
  install          Install packages.
  download         Download packages.
  uninstall        Uninstall packages.
  freeze           Output installed packages in requirements format.
  inspect          Inspect the python environment.
  list             List installed packages.
  show             Show information about installed packages.
  check            Verify installed packages have compatible dependencies.
  config           Manage local and global configuration.
  search           Search PyPI for packages.
  cache            Inspect and manage pip's wheel cache.
  index            Inspect information available from package indexes.
  wheel            Build wheels from your requirements.
  hash             Compute hashes of package archives.
  completion       A helper command used for command completion.
  debug            Show information useful for debugging.
  help             Show help for commands.

General Options:
  -h, --help          Show help.
  --debug             Let unhandled exceptions propagate outside the main subroutine, instead of logging them to stderr.
  --isolated           Run pip in an isolated mode, ignoring environment variables and user configuration.
  --require-virtualenv Allow pip to only run in a virtual environment; exit with an error otherwise.
  --python <python>   Run pip with the specified Python interpreter.
  -v, --verbose        Give more output. Option is additive, and can be used up to 3 times.
  -V, --version        Show version and exit.
  -q, --quiet          Give less output. Option is additive, and can be used up to 3 times (corresponding to WARNING, ERROR, and CRITICAL logging levels).
  --log <path>        Path to a verbose appending log.
  --no-input           Disable prompting for input.
  --keyring-provider <keyring_provider> Enable the credential lookup via the keyring library if user input is allowed. Specify which mechanism to use [disabled, import, subprocess]. (default: disabled)
  --proxy <proxy>     Specify a proxy in the form scheme://[user:passwd@]proxy.server:port.
  --retries <retries> Maximum number of retries each connection should attempt (default 5 times).
  --timeout <sec>     Set the socket timeout (default 15 seconds).
  --exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup, (a)bort.
  --trusted-host <hostname> Mark this host or host:port pair as trusted, even though it does not have valid or any
```

Задание 3

В Replit модуль main.py создан для вызова двух программ из других модулей, т.к. в других случаях модуль task2.py не запускался по неизвестной ошибке.

Main

```
1  '''Модуль для вызова функций из task1.py и task2.py'''
2
3  print('Задача 3.1')
4  import task1
5  print('Задание 3.2')
6  import task2
7
```

Задача 3.1

Введите первое число: 5

Введите второе число: 45

Введите оператор: *

225.0

test passed

test passed

test passed

test passed

Задание 3.2

Введите границы диапазона

Введите начало диапазона: 3

Введите конец диапазона: 89

Введите загаданное число в пределах диапазона: 13

Загаданное число: 13, Число сравнений: 3

3.1

```
1 def calculate(a: float, b: float, operator: str):
2     ''' Выполняет определенную арифметическую операцию над двумя числами.
3     Если заданный операнд равен предложенным, то выполняется соответствующая операция и возвращается результат вычисления.
4     Аргументы:
5     a - первое число.
6     b - второе число.
7     operator - оператор, который задается и используется для выполнения операции.
8     return - возвращает результат выполнения арифметической операции.
9     ...
10    if operator == '+':
11        return a + b
12    elif operator == '-':
13        return a - b
14    elif operator == '*':
15        return a * b
16    elif operator == '/':
17        if b == 0:
18            return 'На ноль делить нельзя!'
19        return a / b
20    else:
21        return 'Недопустимый оператор'
22
23 def test_add():
24     '''Проверяет функцию calculate() на сложения чисел через тестирование.'''
25     assert calculate(1, 2, '+') == 3
26     return 'test passed'
27
28 def test_sub():
29     '''Проверяет функцию calculate() на вычитание чисел через тестирование.'''
30     assert calculate(4, 65, '-') == -61
31     return 'test passed'
32
33 def test_mult():
34     '''Проверяет функцию calculate() на умножение чисел через тестирование.'''
35     assert calculate(5, 5, '*') == 25
36     return 'test passed'
37
38 def test_div():
39     '''Проверяет функцию calculate() на деление чисел через тестирование.'''
40     assert calculate(10, 2, '/') == 5
41     return 'test passed'
42
43 def main():
44     '''Основная функция.
45     Аргументы:
46     number1 - первое число.
47     number2 - второе число.
48     operator - оператор, который задается и используется для выполнения операции.
49     ...
50     number1 = float(input('Введите первое число: '))
51     number2 = float(input('Введите второе число: '))
52     operator = str(input('Введите оператор: '))
53
54     print(calculate(number1, number2, operator))
55     print(test_add())
56     print(test_sub())
57     print(test_div())
58     print(test_mult())
59
60     main()
```

```
Задача 3.1
Введите первое число: 5
Введите второе число: 45
Введите оператор: *
225.0
test passed
test passed
test passed
test passed
```

3.2

```
1  def binary_search():
2      '''Выполняет бинарный поиск по заданным числам.
3      Если число найдено, то возвращает загаданное число и количество итераций.
4      Если не найдено, то приходит функция сообщает об этом.
5      Аргументы:
6      number1 - первое число, начало диапазона.
7      number2 - второе число, конец диапазона.
8      number3 - третье число, загаданное для поиска.
9      counter - параметр цикла, который считает количество итераций.
10     ...
11     print('Введите границы диапозона')
12     number1 = int(input('Введите начало диапазона: '))
13     number2 = int(input('Введите конец диапазона: '))
14     number3 = int(input('Введите загаданное число в пределах диапозона: '))
15     counter = 0
16
17     while number1 <= number2:
18         center = (number1 + number2) // 2
19         counter += 1
20         if number3 == center:
21             print(f'Загаданное число: {center}, Число сравнений: {counter}')
22             return
23         elif number3 > center:
24             number1 = center + 1
25         else:
26             number2 = center - 1
27     print('Загаданное число не входит в диапозон')
28
29     print(f'Загаданное число: {number3}, Число сравнений: {counter}')
30
31     binary_search()
```

1001-раздел

Задание 3.2

Введите границы диапазона

Введите начало диапазона: 3

Введите конец диапазона: 89

Введите загаданное число в пределах диапазона: 13

Загаданное число: 13, Число сравнений: 3