

**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 №7 по дисциплине:  
«Основы программной инженерии»**

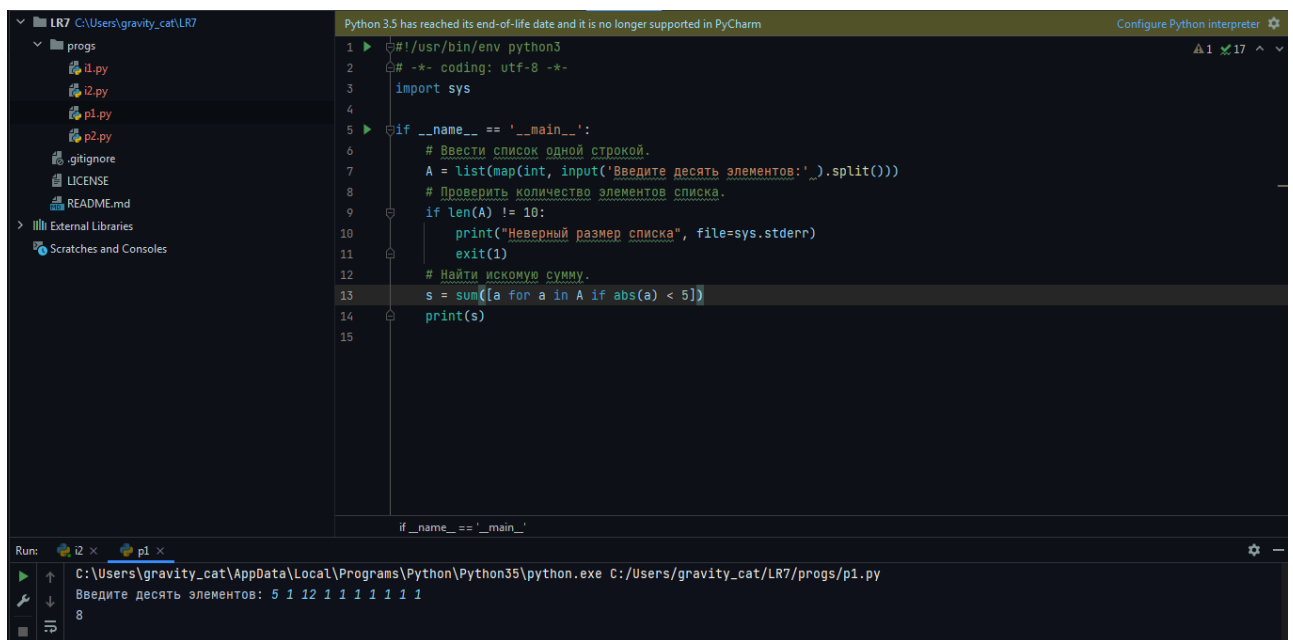
Выполнил:  
Гребен Владислав  
Александрович,  
2 курс, группа ПИЖ-б-о-20-1,

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

Отчет защищен с оценкой\_\_\_\_\_Дата защиты\_\_\_\_\_

Ставрополь, 2021 г.

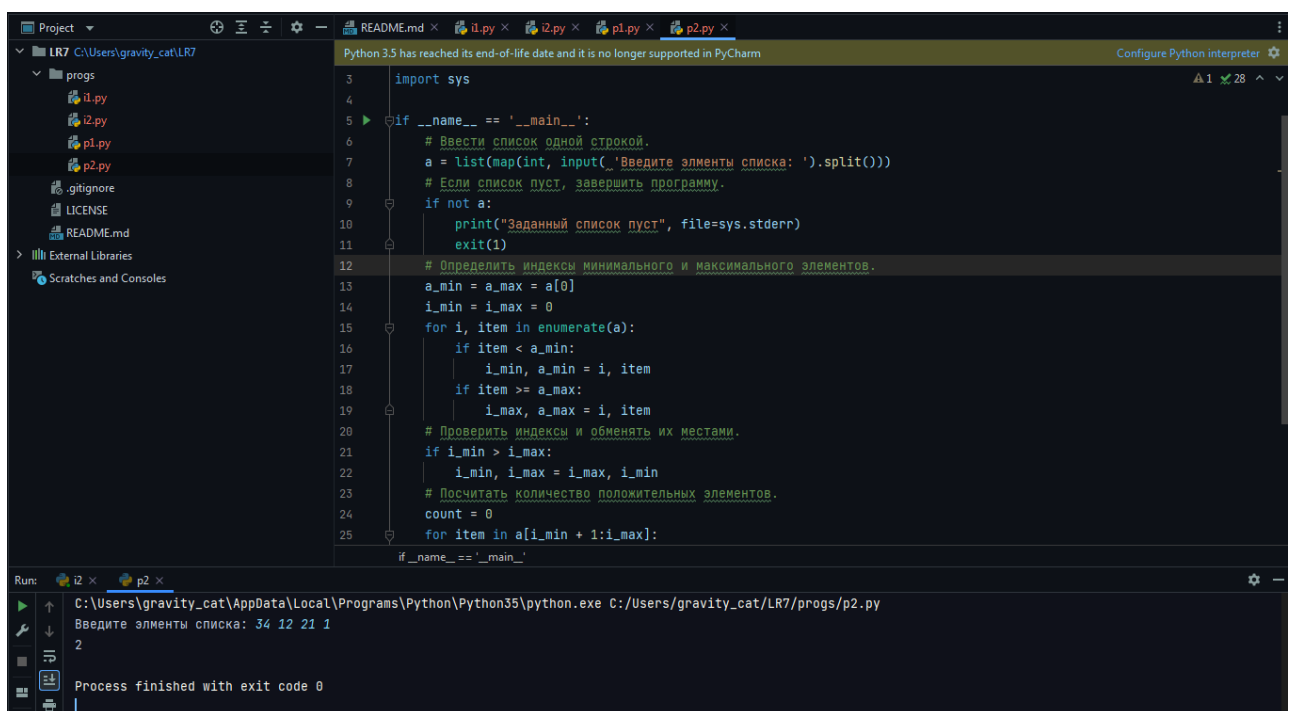
## ВЫПОЛНЕНИЕ:



```
Python 3.5 has reached its end-of-life date and it is no longer supported in PyCharm
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  import sys
4
5  if __name__ == '__main__':
6      # Ввести список одной строкой.
7      A = list(map(int, input('Введите десять элементов: ').split()))
8      # Проверить количество элементов списка.
9      if len(A) != 10:
10         print("Неверный размер списка", file=sys.stderr)
11         exit(1)
12     # Найти искомую сумму.
13     s = sum([a for a in A if abs(a) < 5])
14     print(s)
15
```

Run: C:\Users\gravity\_cat\AppData\Local\Programs\Python\Python35\python.exe C:\Users\gravity\_cat\LR7\progs\p1.py  
Введите десять элементов: 5 1 12 1 1 1 1 1 1 1  
8

Рисунок 7.1 – Первый пример.



```
1  import sys
2
3  if __name__ == '__main__':
4      # Ввести список одной строкой.
5      a = list(map(int, input('Введите элементы списка: ').split()))
6      # Если список пуст, завершить программу.
7      if not a:
8          print("Заданный список пуст", file=sys.stderr)
9          exit(1)
10     # Определить индексы минимального и максимального элементов.
11     a_min = a_max = a[0]
12     i_min = i_max = 0
13     for i, item in enumerate(a):
14         if item < a_min:
15             i_min, a_min = i, item
16         if item >= a_max:
17             i_max, a_max = i, item
18     # Проверить индексы и обменять их местами.
19     if i_min > i_max:
20         i_min, i_max = i_max, i_min
21     # Посчитать количество положительных элементов.
22     count = 0
23     for item in a[i_min + 1:i_max]:
24         if item > 0:
25             count += 1
26
```

Run: C:\Users\gravity\_cat\AppData\Local\Programs\Python\Python35\python.exe C:\Users\gravity\_cat\LR7\progs\p2.py  
Введите элементы списка: 34 12 21 1  
2  
Process finished with exit code 0

Рисунок 7.2 – Второй пример.

# ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

## Вариант 5

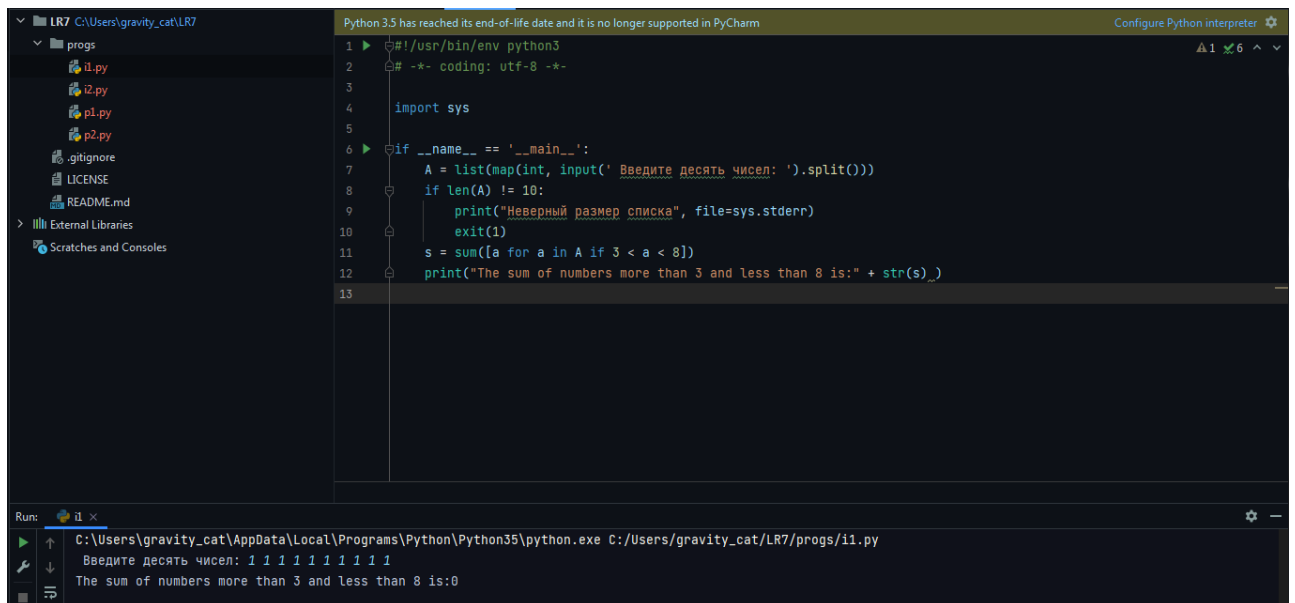


Рисунок 7.3 – Задание №1

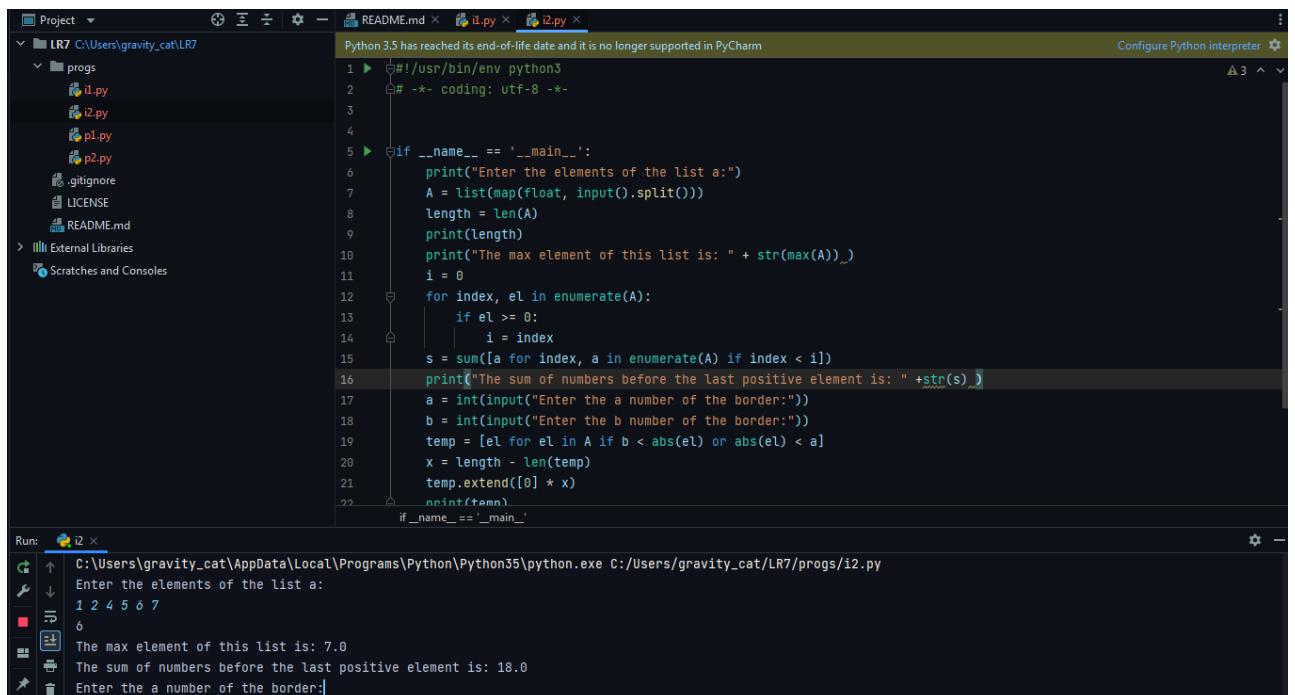


Рисунок 7.4 – Задание №2

## ОТВЕТЫ НА ВОПРОСЫ

1. Список (list) – это структура данных для хранения объектов различных типов.
2. Для создания списка нужно заключить элементы в квадратные скобки.
3. При его создании в памяти резервируется область, которую можно условно назвать некоторым “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от таких типов данных как число или строка, содержимое “контейнера” списка можно менять.
4. Читать элементы списка можно с помощью следующего цикла: `my_list = ['один', 'два', 'три', 'четыре', 'пять'] for elem in my_list: print(elem)`
5. Списки можно объединять с помощью операции сложения, а также повторять с помощью умножения.
6. Нужно использовать оператор «in»
7. Метод count можно использовать для определения числа сколько раз данный элемент встречается в списке.
8. Метод insert можно использовать, чтобы вставить элемент в список: `my_list = [1, 2, 3, 4, 5] my_list.insert(1, 'Привет') print(my_list)` Метод append можно использовать для добавления элемента в список: `my_list = ['один', 'два', 'три', 'четыре', 'пять'] my_list.append('ещё один') print(my_list)`
9. Для сортировки списка нужно использовать метод sort. Пример: `list_2.sort()`.
10. Удалить элемент можно, написав его индекс в методе pop: `removed = my_list.pop(2)` Если не указывать индекс, то функция удалит последний элемент. Элемент можно удалить с помощью метода remove. `my_list.remove('')` Оператор del можно использовать для тех же целей: `del my_list[2] del my_list[1:3]`
11. Пример использования списковых включений для ввода данных в список: `a = [i for i in range(n)]` Использование функции map: `a = [1, 2, 3, 4, 5, 6, 7] b = list(map(lambda x: x**2, a)) for i in a: if i%2 == 0: b.append(i)` Данный алгоритм можно реализовать с помощью функции filter: `a = [1, 2, 3, 4, 5, 6, 7] b = list(filter(lambda x: x % 2 == 0, a))`
12. Слайс задается тройкой чисел, разделенных запятой: start:stop:step. Start – позиция с которой нужно начать выборку, stop – конечная позиция, step – шаг. При этом необходимо помнить, что выборка не включает элемент, определяемый stop.
13. len(), min(), max(), sum().
14. `list2 = my_list` создается ссылка на данный список.
15. sorted() возвращает новый отсортированный список, оставляя исходный список незатронутым. list.sort() сортирует список на месте, изменяет индексы списка и возвращает None (как и все операции на месте). sorted() работает с любыми итерациями, а не только со списками. Строки, кортежи, словари (вы получите ключи), генераторы и т. Д., возвращающие список,

содержащий все элементы, отсортированные.

- Используйте `list.sort()` , когда вы хотите изменить список, `sorted()` , когда вы хотите вернуть новый отсортированный объект.
- Используйте `sorted()` , когда вы хотите отсортировать что-то, что является повторяемым, а не списком .
- Для списков `list.sort()` быстрее, чем `sorted()` , потому что ему не нужно создавать копию. Для любой другой операции у вас нет выбора.