

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития  
Кафедра инфокоммуникаций

**ОТЧЕТ**  
**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.4**  
**дисциплины «Основы кроссплатформенного программирования»**  
**Вариант \_\_\_\_**

Выполнил:  
Медяник Даниил Владимирович  
1 курс, группа ИТС-б-0-22-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А., доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

## Тема: Работа со списками в языке Python

**Цель:** приобретение навыков по работе со списками при написании программ с помощью языка программирования Python версии 3.x

## Ход работы

**Условие примера:** ввести список А из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.

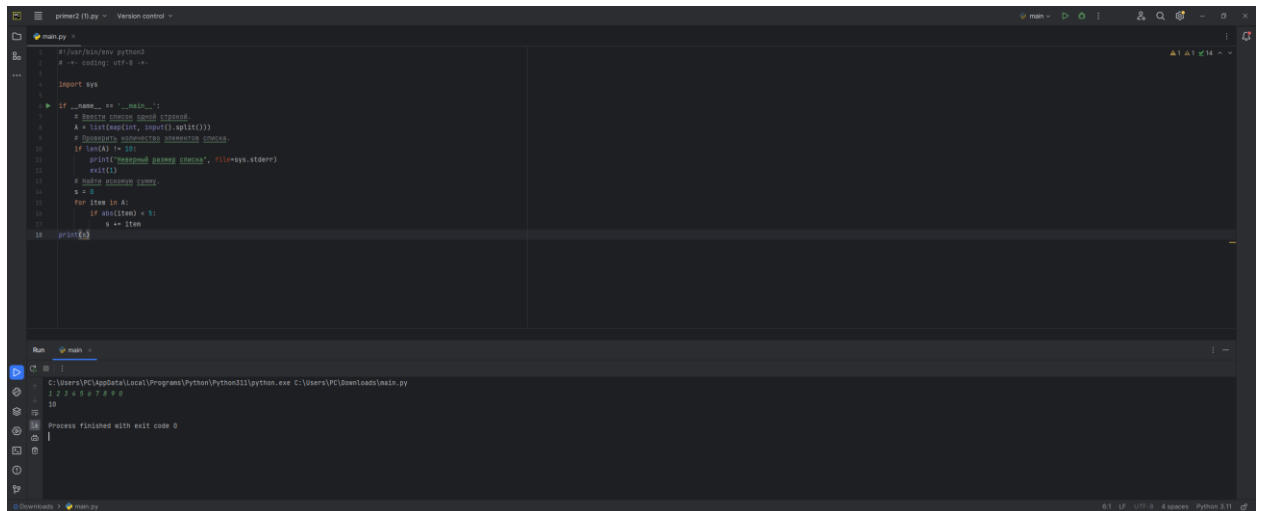


Рисунок 1. Реализация первого примера

**Условие примера:** написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

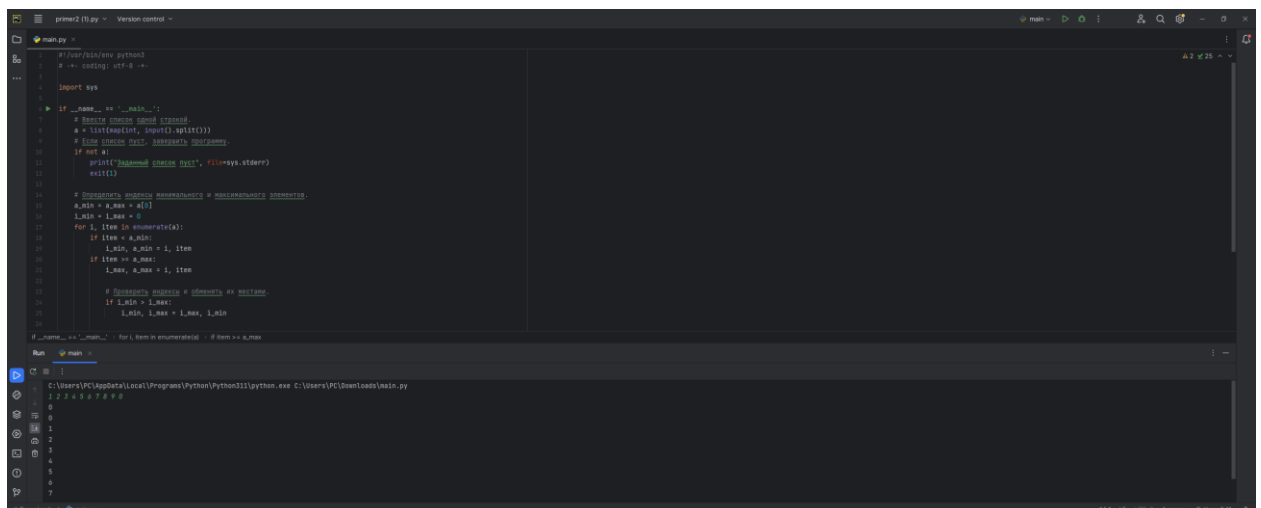
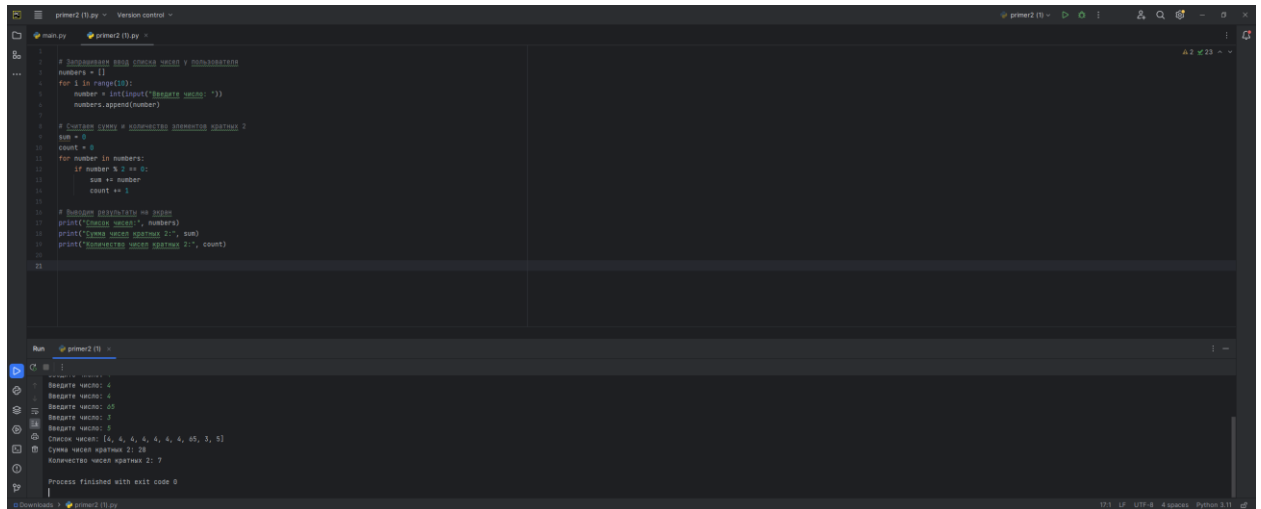


Рисунок 2. Реализация второго примера

## Индивидуальное задание

## Вариант 16

**Условие примера:** Ввести список A из 10 элементов, найти сумму элементов кратных 2, их количество и вывести результаты на экран.



```
1 # Генерируем список чисел через random
2 numbers = []
3 for i in range(10):
4     number = randint(0, randint(0, 100))
5     numbers.append(number)
6
7 # Считаем сумму и количество элементов кратных 2
8 sum = 0
9 count = 0
10 for number in numbers:
11     if number % 2 == 0:
12         sum += number
13         count += 1
14
15 # Выводим результаты на экран
16 print("Список чисел: ", numbers)
17 print("Сумма чисел кратных 2: ", sum)
18 print("Количество чисел кратных 2: ", count)
19
20
```

Run primer2 (1) -

```
Введите число: 4
Введите число: 4
Введите число: 65
Введите число: 7
Введите число: 5
Список чисел: [4, 4, 4, 4, 4, 4, 65, 3, 5]
Сумма чисел кратных 2: 78
Количество чисел кратных 2: 7
Process finished with exit code 0
```

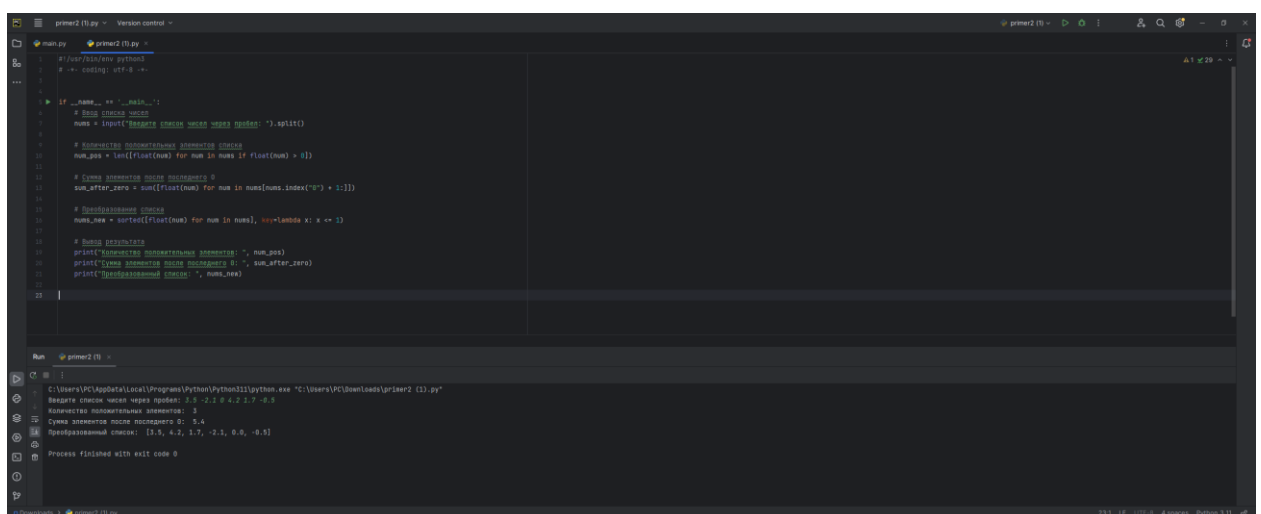
Рисунок 3. Реализация первого примера

**Условие примера:** В списке, состоящем из вещественных элементов, вычислить:

1. количество положительных элементов списка;
2. сумму элементов списка, расположенных после последнего элемента, равного нулю.

Преобразовать список таким образом, чтобы сначала располагались все элементы, целая часть

которых не превышает 1, а потом - все остальные.



```
1 # -*- coding: utf-8 -*-
2
3 if __name__ == '__main__':
4     # Ввод списка чисел
5     nums = input("Введите список чисел через пробел: ").split()
6
7     # Количество положительных элементов списка
8     num_pos = len([float(num) for num in nums if float(num) > 0])
9
10    # Сумма элементов после последнего 0
11    sum_after_zero = sum([float(num) for num in nums[nums.index("0") + 1:]])
12
13    # Преобразование списка
14    nums_new = sorted([float(num) for num in nums], key=lambda x: x // 1)
15
16    # Вывод результатов
17    print("Количество положительных элементов: ", num_pos)
18    print("Сумма элементов после последнего 0: ", sum_after_zero)
19    print("Преобразованный список: ", nums_new)
20
```

Run primer2 (1) -

```
C:\Users\PC\AppData\Local\Programs\Python\Python311\python.exe "C:\Users\PC\Downloads\primer2 (1).py"
Введите список чисел через пробел: 3.5 -2.1 0 4.2 1.7 -0.5
Количество положительных элементов: 5
Сумма элементов после последнего 0: 5.6
Преобразованный список: [1.5, 4.2, 1.7, -2.1, 0.0, -0.5]
Process finished with exit code 0
```

Рисунок 4. Реализация второго примера

## Ответы на контрольные вопросы

1. Что такое списки в языке Python?

Список - это один из встроенных типов данных в языке Python, представляющий собой упорядоченный изменяемый набор объектов произвольных типов. В нем можно хранить объекты различных типов. Переменная, определяемая как список, содержит ссылку на структуру в памяти, которая в свою очередь хранит на какие-либо другие объекты или структуры.

2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки.

3. Как организовано хранение списков в оперативной памяти?

Списки в Python хранятся в оперативной памяти в виде последовательности элементов, каждый из которых может быть любого типа. При создании списка выделяется некоторое количество памяти, которое может быть увеличено или уменьшено в зависимости от изменения размера списка.

4. Каким образом можно перебрать все элементы списка?

Для перебора всех элементов списка можно использовать цикл `for` или метод `for in`:

5. Какие существуют арифметические операции со списками?

Для объединения списков можно использовать оператор сложения ( `+` ). Список можно повторить с помощью оператора умножения ( `*` ).

6. Как проверить есть ли элемент в списке?

Чтобы проверить, содержится ли элемент в списке, можно использовать оператор `in`.

7. Как определить число вхождений заданного элемента в списке?

Чтобы определить число вхождений заданного элемента в списке, можно использовать метод `count()`

8. Как осуществляется добавление (вставка) элемента в список?

Чтобы добавить элемент в список, можно использовать метод `append()`  
Чтобы вставить элемент в список на заданную позицию, можно использовать метод `insert()`

## 9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort`. Для сортировки списка в порядке убывания необходимо вызвать метод `sort` с аргументом `reverse=True`.

## 10. Как удалить один или несколько элементов из списка?

Для удаления одного элемента из списка можно использовать метод `remove()`, указав в скобках значение элемента, который нужно удалить. Удалить элемент можно, написав его индекс в методе `pop`. Если не указывать индекс, то функция удалит последний элемент.

## 11. Что такое списковое включение и как с его помощью осуществлять обработку списков?

В языке Python есть две очень мощные функции для работы с коллекциями: `map` и `filter`. Они позволяют использовать функциональный стиль программирования, не прибегая к помощи циклов, для работы с такими типами как `list`, `tuple`, `set`, `dict` и т.п. Списковое включение позволяет обойтись без этих функций.

## 12. Как осуществляется доступ к элементам списков с помощью срезов?

Доступ к элементам списка с помощью срезов осуществляется с помощью квадратных скобок `[]`. Срезы задаются в виде `start:stop:step`, где `start` - индекс первого элемента в срезе (включительно), `stop` - индекс последнего элемента в срезе (не включительно), а `step` - шаг, с которым нужно выбирать элементы из списка.

## 13. Какие существуют функции агрегации для работы со списками?

1. `len(L)` - получить число элементов в списке `L`
2. `min(L)` - получить минимальный элемент списка `L`
3. `max(L)` - получить максимальный элемент списка `L`
4. `sum(L)` - получить сумму элементов списка `L`, если список `L` содержит только числовые значения.

## 14. Как создать копию списка?

`copy.copy(x)`

15. Самостоятельно изучите функцию `sorted` языка Python. В чем ее отличие от метода `sort` списков?

Функция `sort()` очень похожа на `sorted ()`, но в отличие от `sorted` она ничего не возвращает и не вносит изменений в исходную последовательность. Более того, `sort()` является методом класса `list` и может использоваться только со списками

**Вывод:** приобрел навыки по работе со списками при написании программ с помощью языка программирования Python версии 3.x