

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Северо-Кавказский федеральный университет»**

**Департамент ЦРСиЭ**

**Отчет по лабораторной работе №4  
«Работа со списками и кортежами в языке Python»  
*Вариант 9***

Выполнил студент группы ИВТ-б-о-24-1

Каиров В. С. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

Работа защищена « » \_\_\_\_\_ 20\_\_ г.

Проверил Воронкин Р.А. \_\_\_\_\_  
(подпись)

Ставрополь 2025

### Порядок выполнения работы:

Цели работы: приобретение навыков по работе со списками и кортежами при написании программ с помощью языка программирования Python версии 3.x.

Ссылка на github: <https://github.com/ozetin/labochka4>

В ходе данной лабораторной работы были проработаны приведённые примеры:

Ввести список A из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран.

```
1  #!/usr/bin/env python3
2  -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == '__main__':
7      A = list(map(int, input().split()))
8      if len(A) != 10:
9          print("Неверный размер списка", file=sys.stderr)
10         exit(1)
11
12     s = 0
13     for item in A:
14         if abs(item) < 5:
15             s += item
16     print(s)
```

Рисунок 1. Программа решения примера 1

```
C:\Users\vadim\labochka4\.venv\Scripts>
1 5 7 9 5 4 8 97 4 3
12

Process finished with exit code 0
```

Рисунок 2. Результат работы программы

Рассмотрим еще один способ решения данной задачи с помощью списковых включений:

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = list(map(int, input().split()))
    if len(A) != 10:
        print("Неверный размер списка", file=sys.stderr)
        exit(1)

    s = sum([a for a in A if abs(a) < 5])
    print(s)
```

Рисунок 3. Программа решения примера 1

```
C:\Users\vadim\labochka4\.venv\Script
1 5 7 9 5 4 8 97 6 5
5

Process finished with exit code 0
|
```

Рисунок 4. Результат работы программы

Пример 2. Написать программу, которая для целочисленного списка определяет, сколько положительных элементов располагается между его максимальным и минимальным элементами.

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    a = list(map(int, input().split()))
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    a_min = a_max = a[0]
    i_min = i_max = 0
    for i, item in enumerate(a):
        if item < a_min:
            i_min, a_min = i, item

        if item >= a_max:
            i_max, a_max = i, item

    if i_min > i_max:
        i_min, i_max = i_max, i_min

    count = 0
    for item in a[i_min+1:i_max]:
        if item > 0:
            count += 1
    print(count)
```

Рисунок 5. Программа решения примера 2

```
C:\Users\vadim\labochka4\.venv\
1 2 99 -1 -2 8 7 -99 3 4
2
```

Рисунок 6. Результат программы решения примера 2

Пример 3. Ввести кортеж А из 10 элементов, найти сумму элементов, меньших по модулю 5, и вывести ее на экран. Использовать в программе вместо списков кортежи.

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = tuple(map(int, input().split()))
    if len(A) != 10:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

Рисунок 7. Программа решения примера 3

```
C:\Users\vadim\labochka4\.venv\Scripts>python3
6 7 8 -4 -4 -4 9 10 9 3
-9

Process finished with exit code 0
```

Рисунок 8. Результат программы решения примера 3

Решение данной задачи с помощью списковых включений:

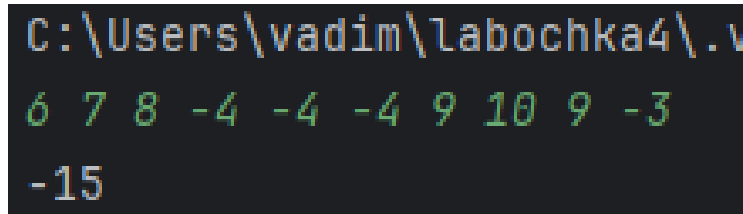
```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    A = tuple(map(int, input().split()))
    if len(A) != 10:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    s = sum(a for a in A if abs(a) < 5)
    print(s)
```

Рисунок 9. Программа решения примера 3



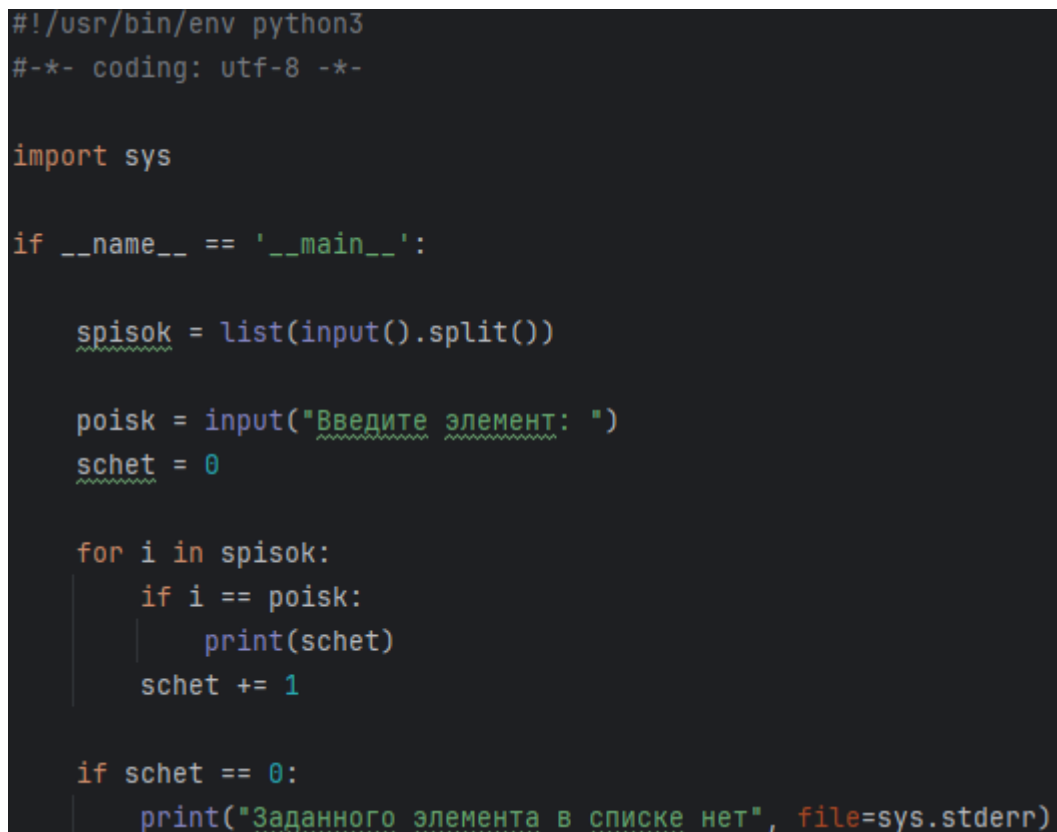
```
C:\Users\vadim\labochka4\.  
6 7 8 -4 -4 -4 9 10 9 -3  
-15
```

Рисунок 10. Результат программы решения примера 3

Далее была проделана работа по выполнению индивидуальных заданий.

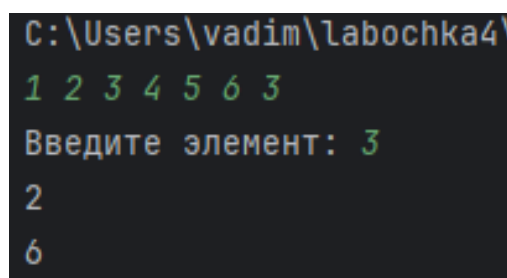
Задание 1. Составить программу с использованием одномерных массивов для решения задачи. Решить как с использованием циклов, так и с использованием List Comprehensions.

Составить программу, выдающую индексы заданного элемента или сообщающую, что такого элемента в списке нет.



```
#!/usr/bin/env python3  
#-*- coding: utf-8 -*-  
  
import sys  
  
if __name__ == '__main__':  
  
    spisok = list(input().split())  
  
    poisk = input("Введите элемент: ")  
    schet = 0  
  
    for i in spisok:  
        if i == poisk:  
            print(schet)  
            schet += 1  
  
    if schet == 0:  
        print("Заданного элемента в списке нет", file=sys.stderr)
```

Рисунок 11. Программа решения задания 1



```
C:\Users\vadim\labochka4\  
1 2 3 4 5 6 3  
Введите элемент: 3  
2  
6
```

Рисунок 12. Результат программы решения задания 1

Решение задания 1 с использованием List Comprehensions:

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    spisok = list(input().split())

    poisk = input("Введите элемент: ")
    naiden = [i for i, elem in enumerate(spisok) if elem == poisk]

    if len(naiden) == 0:
        print("Заданного элемента в списке нет", file=sys.stderr)
    else:
        print(naiden)
```

Рисунок 13. Программа решения задания 1

```
C:\Users\vadim\labochka4\
1 2 3 4 5 6 5 5
Введите элемент: 5
[4, 6, 7]
```

Рисунок 14. Результат программы решения задания 1

Задание 2. В списке, состоящем из целых элементов, вычислить:

- 1) Минимальный по модулю элемент списка
- 2) Сумму модулей элементов списка, расположенных после первого элемента, равного нулю.

Преобразовать список таким образом, чтобы в первой его половине располагались элементы, стоявшие в четных позициях, во второй половине – элементы, стоявшие в нечетных позициях

```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    a = list(map(int, input().split()))
    if not a:
        print("Заданный список пуст", file=sys.stderr)
        exit(1)

    a_min = a_max = a[0]
    i_min = i_max = 0
    for i, item in enumerate(a):
        if item < a_min:
            i_min, a_min = i, item

        if item >= a_max:
            i_max, a_max = i, item

    if i_min > i_max:
        i_min, i_max = i_max, i_min

    count = 0
    for item in a[i_min+1:i_max]:
        if item > 0:
            count += 1

    print(count)
```

Рисунок 15. Программа решения задания 2

```
C:\Users\vadim\labochka4\.venv
1 2 3 4 5 67 8 9
1 В списке нет нуля
[1, 3, 5, 8, 2, 4, 67, 9]
```

Рисунок 16. Результат программы решения задания 2

Задание 3. Если в кортеже есть хотя бы одна тройка соседних чисел, в которой средний элемент больше своих соседей, т.е. предшествующего и последующего, то напечатать все элементы, предшествующие элементам последней из таких троек.



```
#!/usr/bin/env python3
#-*- coding: utf-8 -*-

import sys

if __name__ == '__main__':
    t = tuple(map(int, input().split()))

    index_last = -1

    for i in range(1, len(t) - 1):
        if t[i] > t[i - 1] and t[i] > t[i + 1]:
            index_last = i - 1

    if index_last != -1:
        print(t[:index_last])
    else:
        print("Подходящих троек нет", file=sys.stderr)
```

Рисунок 17. Программа решения задания 3

```
C:\Users\vadim\labochka4\
1 1 1 5 1 1 1 6 1 1 1
(1, 1, 1, 5, 1, 1)
```

Рисунок 18. Результат программы решения задания 3

Контрольные вопросы:

1. Что такое списки в языке Python?

Список – это структура данных для хранения объектов различных типов.

Список похож на массив, только хранит объекты различных типов.

2. Как осуществляется создание списка в Python?

Для создания списка нужно заключить элементы в квадратные скобки:

```
my_list = [1, 2, 3, 4, 5]
```

3. Как организовано хранение списков в оперативной памяти?

При создании массива в памяти резервируется область, называемая “контейнером”, в котором хранятся ссылки на другие элементы данных в памяти. В отличие от чисел и строк содержимое “контейнера” списка можно менять.

4. Каким образом можно перебрать все элементы списка?

```
my_list = [1, 2, 3, 4, 5]
```

```
for elem in my_list:
```

```
    print(elem)
```

5. Какие существуют арифметические операции со списками?

Объединение списков оператором сложения.

Повторить список оператором умножения:

```
List_1 = [1, 2, 3]
```

```
Print(list_1 * 2)
```

Результат:

```
[1, 2, 3, 1, 2, 3]
```

6. Как проверить есть ли элемент в списке?

Чтобы проверить есть ли заданный элемент в списке Python необходимо использовать оператор `in`.

7. Как определить число вхождений заданного элемента в списке?

Метод `count`.

8. Как осуществляется добавление элемента в список?

Методом `insert`, сначала в скобках пишется индекс потом элемент, который мы вставляем.

Также методом `append` можно добавить в конец списка.

9. Как выполнить сортировку списка?

Метод `sort`

10. Как удалить один или несколько элементов из списка?

Можно удалить элемент по индексу методом `pop` или методом `remove`, указав сам элемент. Также можно удалить оператором `del`:

```
del my_list[2]
```

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

Пример:

```
A = [i for i in range(int(input()))]
```

Пример с заменой функции `map`:

```
A = [1, 2, 3, 4, 5, 6, 7]
```

```
B = list(map(lambda x: x**2, a))
```

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

С помощью срезов можно решать задачи выборки элементов из списка.

Срез задаётся тройкой чисел, разделённых двоеточием: start:stop:step.

13. Какие существуют функции агрегации для работы со списками?

len(L) – получить число элементов в списке L

min(L) – получить минимальный элемент списка L

max(L) – получить максимальный элемент списка L

sum(L) – получить сумму элементов списка L

14. Как создать копию списка?

```
a[:]
```

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

Функция sorted() – это универсальный метод сортировки. В качестве обязательного параметра она принимает любой итерируемый объект и возвращает отсортированный список, созданный из его элементов.

Когда используется sorted() исходный объект остается без изменений, а когда метод sort(), исходный объект сортируется.

16. Что такое кортежи в языке Python?

Кортеж – неизменяемая структура данных, которая похожа на список, но его элементы изменять нельзя. Создание кортежа:

```
b = (1, 2, 3)
```

17. Каково назначение кортежей в языке Python?

Причина, по которой используют кортежи – это обезопасить данные от случайного изменения.

18. Как осуществляется создание кортежей?

```
a = ()
```

```
a = tuple()
```

19. Как осуществляется доступ к элементам кортежа?

Доступ к элементам кортежа осуществляется также как в списках через индекс.

20. Зачем нужна распаковка?

Кортежи содержат значения разных типов

21. Какую роль играют кортежи в множественном присваивании?

С помощью кортежей можно обменять значения между двумя переменными:

$a = 100$

$b = \text{'foo'}$

$(a, b) = (b, a)$

22. Как выбрать элементы кортежа с помощью среза?

$T2 = T1[i:j]$

$T2$  – новый кортеж, который получается из кортежа  $T1$

$T1$  – исходный кортеж, для которого происходит срез

$i, j$  – нижняя и верхняя границы среза

23. Как выполняется конкатенация и повторение кортежей?

Конкатенация обозначается символом  $+$

$T3 = T1 + T2$  - конкатенация

$T2 = T1 * n$  - повторение

24. Как выполняется обход элементов кортежа?

Также как и со списками

25. Как проверить принадлежность элемента кортежу?

элемент  $in$  кортеж

26. Какие методы работы с кортежами вам известны?

`Count()`, `index()`.

27. Допустимо ли использование функции агрегации таких как `len()`, `sum()` и т.д. при работе с кортежами?

Да, использование функций агрегации при работе с кортежами полностью допустимо. Кортежи в Python — это итерируемые объекты, а

большинство встроенных функций (таких как `len()`, `sum()`, `min()`, `max()`, `any()`, `all()`) работают с любыми итерируемыми объектами, включая кортежи.

28. Как создать кортеж с помощью спискового включения?

Функцией `tuple()`

Вывод: в ходе данной работы были приобретены навыки работы со списками и кортежами, были изучены их методы и функции.