

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО - КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Кафедра инфокоммуникаций**

**Основы кроссплатформенного программирования**

**Отчет по лабораторной работе №6**

**Работа со словарями в языке Python**

Выполнил студент группы

ИТС-б-о-20-1(2)

Игнатова Е.С. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

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

Проверил доцент

Кафедры инфокоммуникаций, старший  
преподаватель

Воронкин Р.А.

---

(подпись)

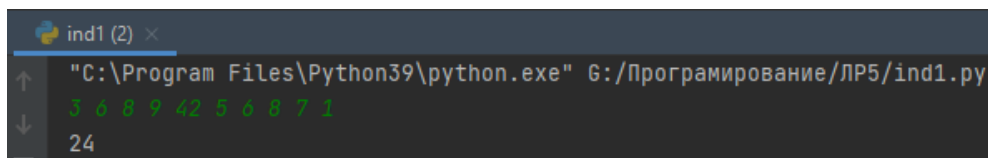
Ставрополь 2021

**Цель работы:** приобретение навыков по работе со словарями при написании программ с помощью языка программирования Python3.

Ссылка на репозиторий: <https://github.com/Nebula139/Sky6>

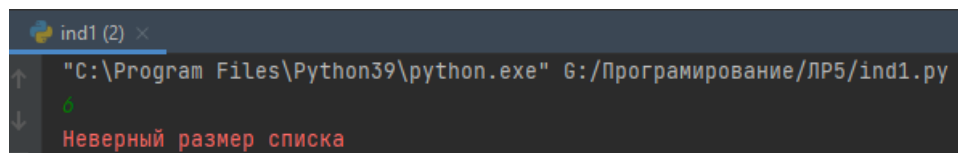
Порядок выполнения работы:

**Пример 1.** Ввести список A из 10 элементов, найти сумму элементов, больших 3 и меньших 8 и вывести ее на экран.



```
ind1 (2) x
"C:\Program Files\Python39\python.exe" G:/Программирование/ЛР5/ind1.py
3 6 8 9 42 5 6 8 7 1
24
```

Рисунок 1. Результат выполнения задания (при вводе списка из 10 элементов)

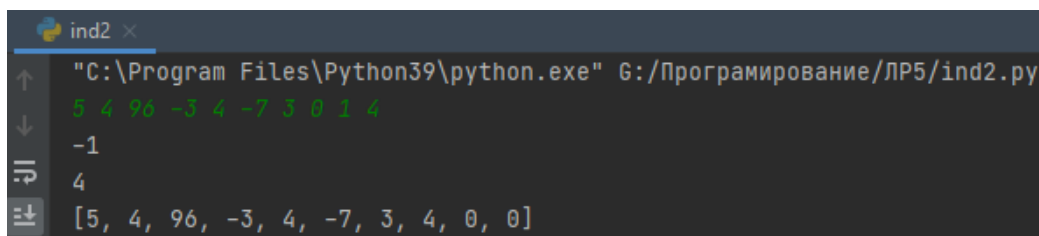


```
ind1 (2) x
"C:\Program Files\Python39\python.exe" G:/Программирование/ЛР5/ind1.py
3
Неверный размер списка
```

Рисунок 2. Результат выполнения задания (при вводе списка из любого другого количества элементов)

**Задание 2.** В списке, состоящем из вещественных элементов, вычислить:

- 1) сумму элементов списка с нечетными номерами;
- 2) сумму элементов списка, расположенных между первым и последним отрицательными элементами.
- 3) Сжать список, удалив из него все элементы, модуль которых не превышает 1. Освободившиеся в конце списка элементы заполнить нулями.



```
ind2 x
"C:\Program Files\Python39\python.exe" G:/Программирование/ЛР5/ind2.py
5 4 96 -3 4 -7 3 0 1 4
-1
4
[5, 4, 96, -3, 4, -7, 3, 4, 0, 0]
```

Рисунок 3. Результат выполнения задания

**Вывод:** были приобретены навыки по работе со списками при написании программ с помощью языка программирования Python3.

**Контрольные вопросы:**

## 1. Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов различных типов. Если вы использовали другие языки программирования, то вам должно быть знакомо понятие массива. Так вот, список очень похож на массив, только, как было уже сказано выше, в нем можно хранить объекты различных типов. Размер списка не статичен, его можно изменять. Список по своей природе является изменяемым типом данных. Переменная, определяемая как список, содержит ссылку на структуру в памяти, которая в свою очередь хранит ссылки 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?

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. Для того, чтобы лучше визуально представлять себе этот процесс взгляните на картинку ниже. Изначально был создан `my_list = [1, 2, 3, 4, 5]` `my_list = ['один', 'два', 'три', 'четыре', 'пять']` `my_list = ['один', 10, 2.25, [5, 15], 'пять']` `third_elem = my_list[2]` список содержащий ссылки на объекты 1 и 2, после операции `a[1] = 3`, вторая ссылка в списке стала указывать на объект 3.

## 4. Каким образом можно перебрать все элементы списка?

Читать элементы списка можно с помощью следующего цикла:

```
my_list = ['один', 'два', 'три', 'четыре', 'пять']  
  
for elem in my_list:  
  
    print(elem)
```

## 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?

Для того, чтобы проверить, есть ли заданный элемент в списке Python необходимо использовать оператор `in`.

#### 7. Как определить число вхождений заданного элемента в списке?

Метод `count` можно использовать для определения числа сколько раз данный элемент встречается в списке

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

- Метод `insert` можно использовать, чтобы вставить элемент в список
- Метод `append` можно использовать для добавления элемента в список

#### 9. Как выполнить сортировку списка?

Для сортировки списка нужно использовать метод `sort`

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

- Удалить элемент можно, написав его индекс в методе `pop`
- Элемент можно удалить с помощью метода `remove`
- Оператор `del` можно использовать для тех же целей

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

List Comprehensions чаще всего на русский язык переводят как абстракция списков или списковое включение, является частью синтаксиса языка, которая предоставляет простой способ построения 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?

Обращение с помощью срезов предполагает указание диапазона в квадратных скобках: `[4:8]`

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

Для работы со списками Python предоставляет следующие функции:

- `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`, если список `L` содержит только числовые значения.

#### 14. Как создать копию списка?

Операция присваивания не копирует объект, он лишь создаёт ссылку на объект. Для изменяемых коллекций, или для коллекций, содержащих изменяемые элементы, часто необходима такая копия, чтобы её можно было изменить, не изменяя оригинал. Данный модуль предоставляет общие (поверхностная и глубокая) операции копирования. `copy.copy(x)`

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

Функция `sorted()` в Python возвращает отсортированный список из элементов в итерируемом объекте. `list.sort()` на 13% быстрее, чем `sorted()`.