

и 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 №7**  
**дисциплины «Программирование на Python»**  
**Вариант 31**

Выполнил:  
Репкин Александр Павлович  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника»,  
направленность (профиль)  
«Программное обеспечение средств  
вычислительной  
техники и автоматизированных  
систем», очная форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р.А., канд. техн. наук,  
доцент кафедры инфокоммуникаций

---

(подпись)

Отчет защищен с оценкой \_\_\_\_\_ Дата защиты \_\_\_\_\_  
Ставрополь, 2023 г.

**Тема:** Работа со списками в языке Python.

**Цель:** приобрести навыки работы со списками при написании программ с помощью языка программирования Python версии 3.x.

**Порядок выполнения работы:**

1. Выполнен первый пример. В нём анализировались 10 введённых чисел. Если по модулю они меньше 5, то они суммировались, после чего сумма выводится на экран.

```
Добрый день! Введите 10 элементов через пробел:-2 -3 2 3 5 8 -8 999 1 0
Сумма элементов - 1
```

Рисунок 1. Полученный результат примера №1.

2. Выполнен второй пример. В нём для целочисленного списка определяется, сколько положительных элементов располагается между его максимальным и минимальным элементами.

```
Добрый день! Введите 10 элементов через пробел:4 -8 -3 -2 -1 6 2 999 0 4
Количество элементов > 0 - 2
```

Рисунок 2. Полученный результат примера №2.

3. Выполнено индивидуальное задание №1 – необходимо найти наименьший элемент и переставить его с последним элементом в введённом списке из 10 элементов.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == '__main__':
8      print("Good day! Please, enter 10 numbers.\n")
9      a = list(map(int, input("Remember to put some space between them:").split()))
10     if len(a) != 10:
11         print("Wrong list size!", file=sys.stderr)
12         exit(1)
13     minimal = sys.maxsize
14     for i in a:
15         if i < minimal:
16             minimal = i
17     if a.count(minimal) > 1:
18         print("List contains several identical minimum values!", file=sys.stderr)
19         exit(1)
20     exchange = a.index(minimal)
21     a[exchange], a[-1] = a[-1], a[exchange]
22     print(a)
23
```

Рисунок 3. Полученный код индивидуального задания №1.

```
Good day! Please, enter 10 numbers.

Remember to put some space between them: 2 3 4 1 9 5 3 2 999 7
New list: [2, 3, 4, 7, 9, 5, 3, 2, 999, 1]
```

Рисунок 4. Полученный результат индивидуального задания №1.

4. Выполнено индивидуальное задание №2 – необходимо в списке, состоящем из вещественных элементов, вычислить: количество элементов списка, лежащих в диапазоне от А до В; сумму элементов списка, расположенных после максимального элемента. Упорядочить элементы списка по убыванию модулей элементов.

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6
7  if __name__ == '__main__':
8      print("Good day! Please, enter 10 numbers.\n")
9      elements = list(
10         map(float, input("Remember to put some space between them:").split()))
11      start = int(input("Please, enter the beginning of the range - "))
12      end = int(input("Please, enter the ending of the range - "))
13      in_range = 0
14      max_element = -1 * sys.maxsize
15      found = False
16      for i in elements:
17         if start <= i <= end:
18             in_range += 1
19         if max_element < i:
20             max_element = i
21      elements_sum = 0
22      for i in elements:
23         if found == True:
24             print(i, end=" ")
25             elements_sum += i
26         elif max_element == i:
27             print("We've found maximal element in list - ",
28                 max_element, " values after it: ", end="")
29             found = True
30      print("\nTheir sum = ", elements_sum)
31      print("\nAmount of elements, that lie in your range is", in_range)
32      elements.sort(key=lambda x: abs(x), reverse=True)
33      print("\nTransformed list: ", elements)
34
```

Рисунок 5. Полученный код индивидуального задания №2.

```
Good day! Please, enter 10 numbers.

Remember to put some space between them:2 -20 3 -5 6 7 3.2 9 1 4
Please, enter the beginning of the range - 3
Please, enter the ending of the range - 8
We've found maximal element in list - 9.0 values after it: 1.0 4.0
Their sum = 5.0

Amount of elements, that lie in your range is 5

Transformed list: [-20.0, 9.0, 7.0, 6.0, -5.0, 4.0, 3.2, 3.0, 2.0, 1.0]
```

Рисунок 6. Полученный результат индивидуального задания №2.

5. Ответы на вопросы.

1) Что такое списки в языке Python?

**Ответ:** списки – структура данных для хранения объектов 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?

**Ответ:** для создания списка нужно заключить элементы в квадратные скобки. Пример: `my_list = ['один', 10, 2.25, [5, 15], 'пять']`.

3) Как организовано хранение списков в оперативной памяти?

**Ответ:** список является изменяемым типом данных. 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) Каким образом можно перебрать все элементы списка?

**Ответ:** читать элементы списка можно с помощью цикла `for` и оператора `in`, например: `for i in list`. Также, можно проходить по индексам списка: `for I in range(0, len(list)): list[i]`.

5) Какие существуют арифметические операции со списками?

**Ответ:** списки можно объединять (`list1 + list2`) и повторять (`list * число повторений`).

6) Как проверить есть ли элемент в списке?

**Ответ:** оператор `in` позволяет проверить наличие элемента в списке.

7) Как определить число вхождений заданного элемента в списке?

**Ответ:** функция `count()` возвращает число повторений требуемого элемента в переданном списке.

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

**Ответ:** метод `insert(индекс, “элемент”)` позволяет вставить требуемый элемент в список по указанному индексу, сдвигая последующие элементы вправо. Если нужно просто добавить элемент в конец списка, то нужен метод `append()`.

9) Как выполнить сортировку списка?

**Ответ:** для сортировки списка есть метод `sort()`, которому можно передать ключ(по какому принципу будет идти сортировка), и параметр `reverse` (Если `True`, то список будет задом-наперёд).

10) Как удалить один или несколько элементов из списка?

**Ответ:** метод `pop(индекс)` позволяет удалить элемент в списке. Очистка списка – метод `clear()`.

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

**Ответ:** списковое включение является частью синтаксиса языка, которая предоставляет простой способ построения списков. Благодаря ему не нужно писать отдельные циклы для заполнения списков, достаточно написать что-то наподобие `a = [i for i in range(n)]`.

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

**Ответ:** слайсы (срезы) являются очень мощной составляющей Python, которая позволяет быстро и лаконично решать задачи выборки элементов из списка. 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`. Пример: `list[1:8:2]`.

13) Какие существуют функции агрегации для работы со списками?

**Ответ:** 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?

**Ответ:** для создания копии списка есть метод `copy`. Также, можно использовать оператор среза.

**15)** В чем отличие функции `sorted` от метода `sort` списков?

**Ответ:** функция `sorted(list)` создаёт новый отсортированный список, не меняя старый, в то время как `sort` изменяет сам переданный ей список.

**Вывод:** в ходе выполнения лабораторной работы были приобретены навыки работы со списками при написании программ с помощью языка программирования Python версии 3.x.