

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»**

Выполнила:  
Кубанова Ксения Олеговна  
2 курс, группа ИВТ-б-о-22-1,  
09.03.01 «Информатика и  
вычислительная техника», очная  
форма обучения

---

(подпись)

Руководитель практики:  
Воронкин Р. А.

---

(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 3.x.

### Порядок выполнения работы

#### Пример 1.

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

**Код:**

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3
4  import sys
5
6  if __name__ == "__main__":
7      #Ввести список одной строкой.
8      A = list(map(int, input().split()))
9      #Проверить кол-во элементов списка.
10     if len(A) != 10:
11         print("Неверный размер списка",
12             exit(1)
13
14     #Найти искомую сумму
15     s = 0
16     for item in A:
17         if abs(item)<5:
18             s += item
19     print(s)
```

Рисунок 1 – код примера 1

**Вывод:**

```
1 5 3 8 9 2 6 8 9 8
5
```

Рисунок 2 – вывод примера 1

Решение предоставлено в файле *prim1.py*.

#### Пример 2.

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

**Код:**

```
prim2.py > ...
5
6  if __name__=='__main__':
7      # ввсети список одной строкой
8      a=list(map(int, input().split()))
9      if not a:
10         print("заданный список пуст", file=sys.stderr)
11         exit(1)
12     a_min=a_max=a[0]
13     i_min=i_max=0
14     for i, item in enumerate(a):
15         if item<a_min:
16             i_min, a_min = i, item
17         if item >= a_max:
18             i_max, a_max= i, item
19     if i_min > i_max:
20         i_min, i_max = i_max, i_min
21     count=0
22     for item in a[i_min+1:i_max]:
23         if item > 0:
24             count +=1
25     print(count)
```

Рисунок 3 – код примера 2

**Вывод:**

```
1 6 4 3 9
3
```

Рисунок 4 – вывод примера 2

Решение примера предоставлено в файле *prim2.py*.

### **Индивидуальное задание 1.**

Ввести список А из 10 элементов, найти разность положительных элементов, и вывести ее на экран.

**Код:**

```

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         r=0
12         for item in A:
13             if item>0:
14                 r-=item
15
16         print("Разноость положительных элементов в списке: ",r)

```

Рисунок 5 – код индивидуального задания 1 (в6)

**Вывод:**

```

6 4 2 -1 8 -3 7 -9 2 -8
Разноость положительных элементов в списке: -29

```

Рисунок 6 – вывод индивидуального задания 1 (в6)

Решение индивидуального задания предоставлено в файле *ind1.py*.

## Индивидуальное задание 2.

В6: В списке, состоящем из целых элементов, вычислить:

- 1) номер максимального элемента списка;
- 2) произведение элементов списка, расположенных между первым и вторым нулевыми элементами.

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

**Код:**

```

if __name__ == '__main__':
    a = list(map(int, input().split()))
    if not a:
        print("заданный список пуст", file = sys.stderr)
        exit(1)
    # поиск индекса максимального элемента
    a_max = a[0]
    i_max = 0
    for i, item in enumerate(a):
        if item >= a_max:
            i_max, a_max = i, item
    print("Номер максимального элемента списка: ", i_max)
    # поиск индексов нулевых элементов для произведения между ними
    p = 1
    z1 = a.index(0)
    z2 = a.index(0, z1+1)
    for j in a[z1+1:z2]:
        p*=j
    print("Произведение элементов между нулевыми значениями: ", p)
    # изменение положения индексов на четные и нечетные
    nch = a[::2]
    ch = a[1::2]
    ar = nch + ch
    print("Список с измененным порядком индексов (сначала нечетные, а после четные): ", ar)

```

Рисунок 7 – код индивидуального задания 2 (в6)

## Вывод:

```

3 5 6 9 0 2 5 3 0 9
Номер максимального элемента списка: 9
Произведение элементов между нулевыми значениями: 30
Список с измененным порядком индексов (сначала нечетные, а после четные): [3, 6, 0, 5, 0, 5, 9, 2, 3, 9]

```

Рисунок 8 – вывод индивидуального задания 2 (в6)

Решение индивидуального задания предоставлено в файле *ind2.py*.

## Контрольные вопросы

### 1 Что такое списки в языке Python?

Список (list) – это структура данных для хранения объектов 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?

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?**

Для считывания элементов в списке можно применить конструкцию:

```
for element in list:  
    print(element)
```

Для обновления элементов существует немного другая конструкция:

```
for element in range(len(my_list)):  
    my_list[element] += 5  
    print(my_list)
```

#### **5 Какие существуют арифметические операции со списками?**

Объединение (+) и повторение (\*).

#### **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.

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

List.insert(<элемент после которого будет вставка>, ‘вставка’)

List.append(‘добавление в конец списка’)

#### **9 Как выполнить сортировку списка?**

List.sort()

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

List.pop(‘индекс удаляемого элемента’)

List.clear() – удаляет все элементы списка

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

Списковое включение – это упрощенный способ создавать и обрабатывать списки. Обработка списков осуществляется с помощью функций filter и map.

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

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 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 , если список L содержит только числовые значения.

## **14 Как создать копию списка?**

```
New_list = old_list.copy()
```

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

Основное различие между ними заключается в том, что list.sort () сортирует список на месте, изменяя его индексы и возвращая None, тогда как sorted () возвращает новый отсортированный список, оставляя исходный список неизменным.

**Вывод:** в ходе выполнения лабораторно работы были приобретены навыки по работе со списками при написании программ с помощью языка программирования python 3.x.