

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 ПРАКТИЧЕСКОЙ РАБОТЕ №2.5**  
**дисциплины «Программирование на Python»**  
Вариант №2

Выполнила:  
Беседина Инга 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., канд. технических  
наук, доцент, доцент кафедры  
инфокоммуникаций

---

(подпись)

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

Ставрополь, 2023 г.

**Тема:** Работа с кортежами в языке Python

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

### Ход работы

**Пример 1.** Ввести кортеж A 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__':
    # Ввести кортеж одной строкой.
    A = tuple(map(int, input().split()))
    # Проверить количество элементов кортежа.
    if len(A) != 10:
        print("Неверный размер кортежа", file=sys.stderr)
        exit(1)

    # Найти искомую сумму.
    s = 0
    for item in A:
        if abs(item) < 5:
            s += item

    print(s)
```

```
1 -6 -4 4 3 -7 8 9 12 -5
4
```

Рисунок 1. Результат работы программы

Решение задачи с использованием списковых включений:

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

import sys

if __name__ == '__main__':
```

```
# Ввести кортеж одной строкой.
A = tuple(map(int, input().split()))
# Проверить количество элементов кортежа.
if len(A) != 10:
    print("Неверный размер кортежа", file=sys.stderr)
    exit(1)

# Найти искомую сумму.
s = sum(a for a in A if abs(a) < 5)
print(s)
```

```
13 4 1 -2 4 -5 6 11 -10 3
10
```

Рисунок 2. Результат работы программы

2. В начале кортежа записано несколько равных между собой элементов. Определить количество таких элементов и вывести все элементы, следующие за последним из них. Условный оператор не использовать.

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

if __name__ == '__main__':
    A = tuple(map(int, input().split()))

    k = A.count(A[0])
    print("Количество равных элементов:", k)
    print("Элементы, следующие за равными элементами: ", A[k:len(A)+1])
```

```
6 6 6 6 3 -1 -5 7 2
Количество равных элементов: 4
Элементы, следующие за равными элементами: (3, -1, -5, 7, 2)
```

Рисунок 3. Результат работы программы

### Контрольные вопросы:

1. Списки в языке Python - это упорядоченные изменяемые коллекции объектов. Они могут содержать элементы различных типов данных, и их размер может изменяться

2. Кортежи в языке Python используются для хранения неизменяемых коллекций объектов. Они могут содержать элементы различных типов данных, и их размер также может изменяться

3. Создание кортежей осуществляется путем перечисления элементов в круглых скобках, разделенных запятыми, например: `my_tuple = (1, 2, 3)`

4. Доступ к элементам кортежа осуществляется по индексу, начиная с 0, например: `my_tuple[0]` вернет первый элемент кортежа

5. Распаковка (деструктуризация) кортежа позволяет присвоить значения элементов кортежа переменным, например: `a, b, c = my_tuple`

6. Кортежи играют важную роль в множественном присваивании, когда нужно одновременно присвоить несколько значений переменным

7. Для выбора элементов кортежа с помощью среза используется следующий синтаксис: `my_tuple[start:end:step]`

8. Конкатенация кортежей выполняется с помощью оператора "+", а повторение - с помощью оператора "\*"

9. Обход элементов кортежа можно осуществить с помощью цикла `for`, например: `for item in my_tuple: print(item)`

10. Принадлежность элемента кортежу можно проверить с помощью оператора "in", например: `if item in my_tuple: do something`

11. Некоторые методы работы с кортежами включают `count()`, `index()` и `len()`

12. Да, функции агрегации такие как `len()`, `sum()` и т.д. допустимо использовать при работе с кортежами

13. Кортеж можно создать с помощью спискового включения, например: `my_tuple = tuple(x for x in range(5))`

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

