

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

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

**Кафедра инфокоммуникаций**

**Отчет по лабораторной работе №8**

**по дисциплине «Основы программной инженерии»**

Выполнил студент группы ПИЖ-б-о-20-1

Ваньянц И.М. « » \_\_\_\_\_ 20\_\_ г.

Подпись студента \_\_\_\_\_

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

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

Ставрополь 2021

## ХОД РАБОТЫ

### 1. Пример 1

```
#!/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 1 – Код программы

```
1 2 3 4 5 6 7 8 9 10
10
```

Рисунок 2 – Результат программы

### 2. Индивидуальное задание

Если в кортеже есть хотя бы одна пара одинаковых соседних элементов, то напечатать все элементы, следующие за элементами первой из таких пар.

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

if __name__ == '__main__':
    a = tuple(map(int, input().split()))
    i = 0
    for index, el in enumerate(a):
        k = a.count(el)
        if k >= 2:
            if len(a) > index + 1:
                if a[index + 1] == el:
                    i = index + 1
                    break
    if i:
        i += 1
        print(a[i:])
    else:
        print("Не найдено пар одинаковых соседних элементов!")
```

Рисунок 3 – Код программы

```
1 2 3 5 5 6 7 8 9 10
(6, 7, 8, 9, 10)
```

Рисунок 4 – Пример работы команды

## ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое кортежи в языке Python?  
Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.
2. Каково назначение кортежей в языке Python?  
Существует несколько причин, по которым стоит использовать кортежи вместо списков. Одна из них – это обезопасить данные от случайного изменения. Если мы получили откуда-то массив данных, и у нас есть желание поработать с ним, но при этом непосредственно менять данные мы не собираемся, тогда, это как раз тот случай, когда кортежи придутся как нельзя кстати. Используя их в данной задаче, мы дополнительно получаем сразу несколько бонусов – во-первых, это экономия места. Дело в том, что кортежи в памяти занимают меньший объем по сравнению со списками. Во-вторых – прирост производительности, который связан с тем, что кортежи работают быстрее, чем списки (т. е. на операции перебора элементов и т. п. будет тратиться меньше времени). Важно также отметить, что кортежи можно использовать в качестве ключа у словаря.
3. Как осуществляется создание кортежей?  
Для создания пустого кортежа можно воспользоваться одной из следующих команд: a = () b = tuple() Кортеж с заданным содержанием создается также как список, только вместо квадратных скобок используются круглые.
4. Как осуществляется доступ к элементам кортежа?  
Доступ к элементам кортежа осуществляется также как к элементам списка – через

- указание индекса. `a = (1, 2, 3, 4, 5) print(a[0])`
5. Зачем нужна распаковка (деструктуризация) кортежа?  
Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто. Но есть способ лучше! Как мы кортеж собираем, так его можно и разобрать: `name_and_age = ('Bob', 42) (name, age) = name_and_age name # 'Bob' age # 42`
  6. Какую роль играют кортежи в множественном присваивании?  
Используя множественное присваивание, можно провернуть интересный трюк: обмен значениями между двумя переменными. `(a, b) = (b, a)`
  7. Как выбрать элементы кортежа с помощью среза?  
Общая форма операции взятия среза для кортежа следующая: `T2 = T1[i:j]` `T2` — новый кортеж, который получается из кортежа `T1`; `T1` — исходный кортеж, для которого происходит срез; `i, j` — соответственно нижняя и верхняя границы среза. Фактически берутся ко вниманию элементы, лежащие на позициях `i, i+1, ..., j-1`. Значение `j` определяет позицию за последним элементом среза.
  8. Как выполняется конкатенация и повторение кортежей?  
`T3 = T1 + T2` — Конкатенация `T2 = T1 * n` — Повторение
  9. Как выполняется обход элементов кортежа?  
Элементы кортежа можно последовательно просмотреть с помощью операторов цикла `while` или `for`.
  10. Как проверить принадлежность элемента кортежу?  
С помощью оператора «`in`»
  11. Какие методы работы с кортежами Вам известны?  
`index()` и `count()`
  12. Допустимо ли использование функций агрегации таких как `len()`, `sum()` и т. д. при работе с кортежами?  
Да, т.к они не изменяют элементы в самом кортеже.
  13. Как создать кортеж с помощью спискового включения.  
`A = tuple(map(int, input().split()))`