

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ**

**Отчет о лабораторной работе №5 по дисциплине основы программной
инженерии**

Выполнил:
Выходцев Егор Дмитриевич,
2 курс, группа ПИЖ-б-о-20-1,

Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

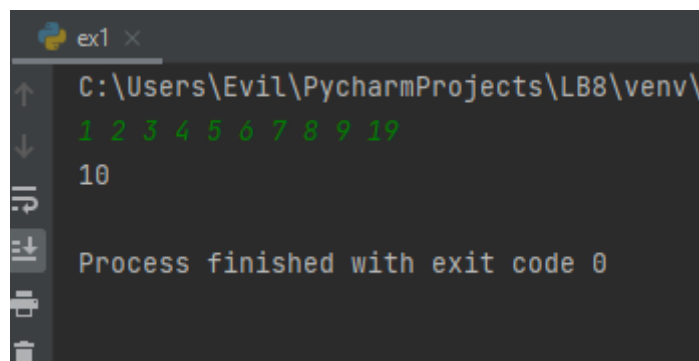
Ставрополь, 2021 г

1. Работа с кортежами в языке Python/

1.1 Пример 1 (рис. 1, 2).

```
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 = tuple(map(int, input().split()))
9     # Проверить количество элементов списка.
10    if len(A) != 10:
11        print("Неверный размер списка", file=sys.stderr)
12        exit(1)
13    # Найти искомую сумму.
14    s = sum(a for a in A if abs(a) < 5)
15    print(s)
16
```

Рисунок 1 – Код примера



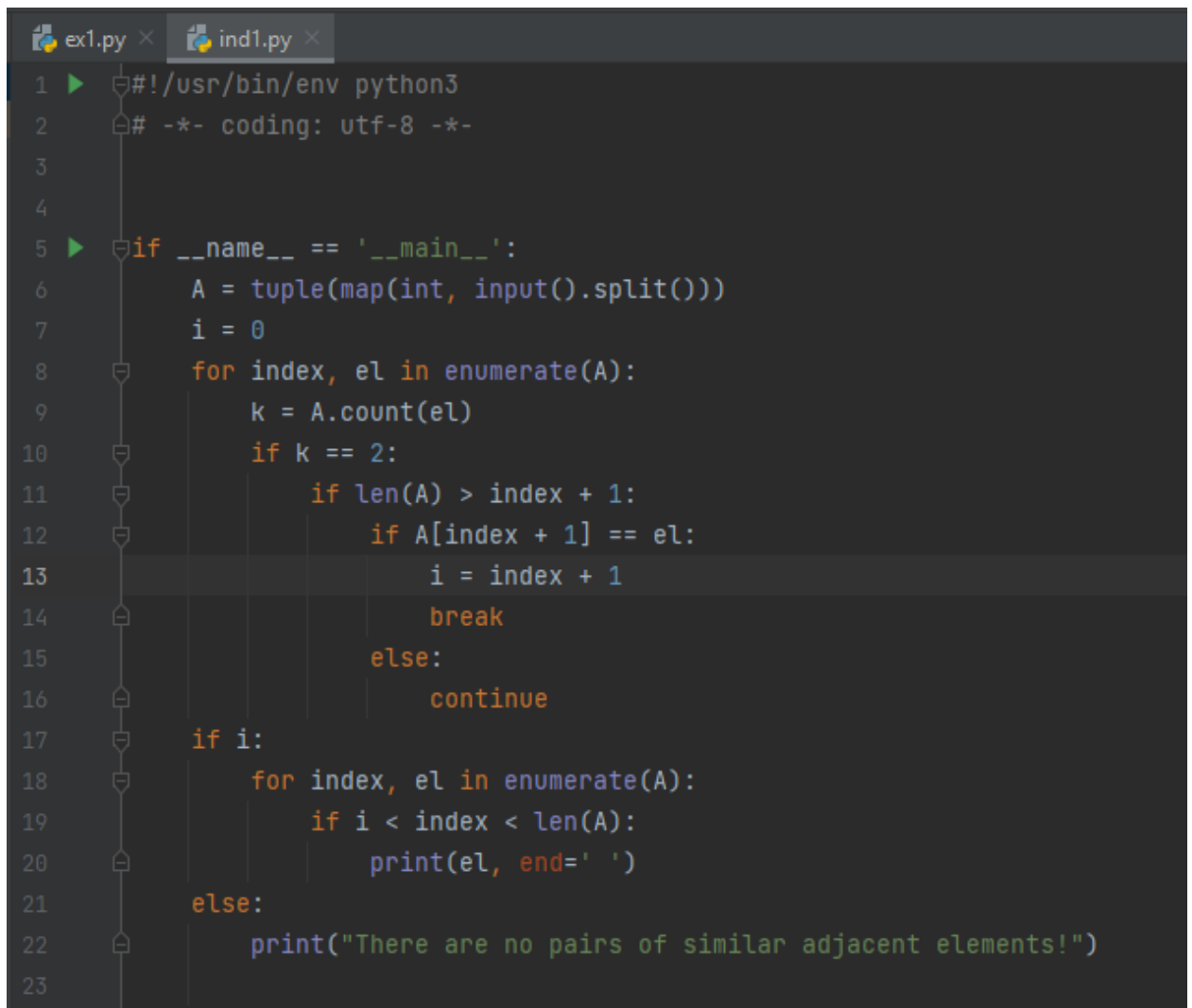
```
ex1 x
C:\Users\Evil\PycharmProjects\LB8\venv\
1 2 3 4 5 6 7 8 9 19
10
Process finished with exit code 0
```

Рисунок 2 – Пример работы программы

Решение индивидуальных заданий

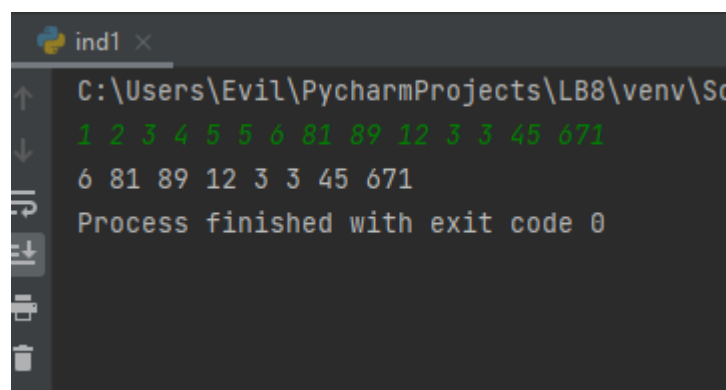
Вариант 5

1.2 Индивидуальное задание №1 (рис. 3



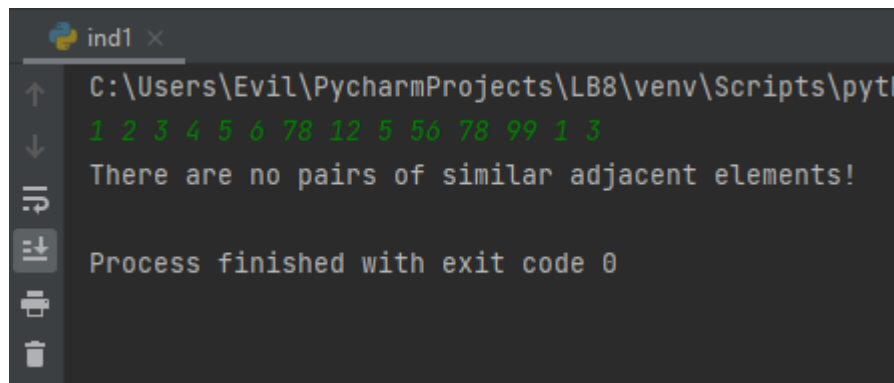
```
1  ▶  #!/usr/bin/env python3
2      # -*- coding: utf-8 -*-
3
4
5  ▶  if __name__ == '__main__':
6      A = tuple(map(int, input().split()))
7      i = 0
8      for index, el in enumerate(A):
9          k = A.count(el)
10         if k == 2:
11             if len(A) > index + 1:
12                 if A[index + 1] == el:
13                     i = index + 1
14                     break
15             else:
16                 continue
17         if i:
18             for index, el in enumerate(A):
19                 if i < index < len(A):
20                     print(el, end=' ')
21         else:
22             print("There are no pairs of similar adjacent elements!")
23
```

Рисунок 3 – Код программы



```
ind1 ×
C:\Users\Evil\PycharmProjects\LB8\venv\Sc
1 2 3 4 5 5 6 81 89 12 3 3 45 671
6 81 89 12 3 3 45 671
Process finished with exit code 0
```

Рисунок 4 – Пример работы программы при наличии двух пар соседних элементов

A screenshot of a terminal window with a dark background. The title bar shows 'ind1' and a close button. The terminal displays the following text: a file path 'C:\Users\Evil\PycharmProjects\LB8\venv\Scripts\pytl', a list of numbers '1 2 3 4 5 6 78 12 5 56 78 99 1 3' in green, the message 'There are no pairs of similar adjacent elements!', and 'Process finished with exit code 0'. On the left side of the terminal, there is a vertical toolbar with icons for navigation and editing.

```
ind1 x
C:\Users\Evil\PycharmProjects\LB8\venv\Scripts\pytl
1 2 3 4 5 6 78 12 5 56 78 99 1 3
There are no pairs of similar adjacent elements!
Process finished with exit code 0
```

Рисунок 5 – Вывод программы при наличии пар одинаковых, но НЕ СОСЕДНИХ элементов

2. 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?

$$T3 = T1 + T2 \text{ – Конкатенация}$$

$$T2 = T1 * n \text{ – Повторение}$$

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.

$$A = \text{tuple}(\text{map}(\text{int}, \text{input}().\text{split}()))$$