

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 №2.5**  
**дисциплины «Введение в специальность»**  
**Вариант \_\_\_\_**

Выполнил:  
Эсеналиев Арсен Мурадинович  
1 курс, группа ИТС-б-о-21-1,  
11.03.02 «Инфокоммуникационные  
технологии и системы связи»,  
направленность (профиль)  
«Инфокоммуникационные системы и  
сети», очная форма обучения

---

(подпись)

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

Ставрополь, 2021 г.

Порядок выполнения работы:

Создаем репозиторий в соответствии с моделью ветвления git-flow и требованиям в задании


## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---

Owner \*

Repository name \*

 Arsen445 ▾


 / 

LB2.5 ✓


Great repository names are short and memorable. Need inspiration? How about [shiny-telegram](#)?

Description (optional)

---

☒  **Public**

Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

---

**Initialize this repository with:**  
Skip this step if you're importing an existing repository.


☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

**Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)


.gitignore template: Python ▾

**Choose a license**  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▾

This will set  **main** as the default branch. Change the default name in your [settings](#).

---

 You are creating a public repository in your personal account.

---

Create repository

```
Cloning into 'LB2.5'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (5/5), done.

D:\>LB2.5
"LB2.5" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

D:\>cd LB2.5

D:\LB2.5>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [D:/LB2.5/.git/hooks]

D:\LB2.5>
```

## Задание 1

Определить, есть ли в кортеже хотя бы одна тройка соседних чисел, в которой средний элемент больше своих «соседей», т. е. предшествующего и последующего. В случае положительного ответа определить номера элементов первой из таких троек.

```

1  #!/usr/bin/env python3
2  # -- coding: utf-8 --
3
4  import math
5
6
7  if __name__ == '__main__':
8      t = []
9      n = int(input('Введите длину кортежа: '))
10     for i in range(n):
11         t.append(int(input()))
12     t = tuple(t)
13     k = 0
14     for i in range(len(t)-2):
15         print(t[i])
16         if t[i+1] > t[i] and t[i+1] > t[i+2]:
17             k += 1
18             print('Есть, его номер(начинается с 1) = ', i+1, i+2, i+3)
19             break
20     if k == 0:
21         print('Нет')

```

```

Введите длину кортежа: 9
7
8
7
4
6
0
6
8
5
7
Есть, его номер(начинается с 1) =  1 2 3

```

## Сохранение изменений и сращивание веток

```

D:\LB2.5>git add .
D:\LB2.5>git commit -m"ghj"
[develop 061ac25] ghj
2 files changed, 61 insertions(+), 3 deletions(-)
create mode 100644 ind/1.py
D:\LB2.5>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
D:\LB2.5>git merge develop
Updating 42de2fd..061ac25
Fast-forward
 .gitignore | 43 ++++++
 ind/1.py   | 21 ++++++
2 files changed, 61 insertions(+), 3 deletions(-)
create mode 100644 ind/1.py
D:\LB2.5>

```

Ответы на контрольные вопросы:

### **Что такое кортежи в языке Python?**

Кортеж (tuple) – это неизменяемая структура данных, которая по своему подобию очень похожа на список.

### **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. Кортежи в памяти занимают меньший объем по сравнению со списками. Кортежи работают быстрее, чем списки

### **Как осуществляется создание кортежей? `a = ()`**

`b = tuple()`

### **Как осуществляется доступ к элементам кортежа?**

Доступ к элементам кортежа осуществляется также как к элементам списка – через указание индекса.

### **Зачем нужна распаковка (деструктуризация) кортежа?**

Обращение по индексу, это не самый удобный способ работы с кортежами. Дело в том, что кортежи часто содержат значения разных типов, и помнить, по какому индексу что лежит — очень непросто.

### **Какую роль играют кортежи в множественном присваивании?**

Используя множественное присваивание, можно проверить интересный трюк: обмен значениями между двумя переменными.

### **Как выбрать элементы кортежа с помощью среза?**

С помощью операции взятия среза можно получить другой кортеж.

Общая форма операции взятия среза для кортежа следующая

`T2 = T1[i:j]` здесь

- `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.

### **Как выполняется конкатенация и повторение кортежей?**

Для кортежей можно выполнять операцию конкатенации, которая обозначается символом +.  $T3 = T1 + T2$

### **Как выполняется обход элементов кортежа?**

Элементы кортежа можно последовательно просмотреть с помощью операторов цикла while или for.

### **Как проверить принадлежность элемента кортежу?**

Проверка вхождения элемента в кортеж - оператор in.

**Какие методы работы с кортежами Вам известны?** index(), count().

**Допустимо ли использование функций агрегации таких как len(), sum() и т. д. при работе с кортежами?**

Доступно.

**Как создать кортеж с помощью спискового включения.**

Так же как и список.