



## Лабораторная работа № 2

Перед началом выполнения каждой лабораторной работы, необходимо выполнить код в двух ячейках ниже. В случае если модуль `datetime` не установлен нужно установить. В последнем принте, вывести свою фамилию и инициалы.

In [1]:

```
!whoami
```

```
laptop-sl701val\home
```

In [2]:

```
from datetime import datetime

current_time = datetime.now()
print(current_time)
print("Rogozhina N.A.") # написать здесь свою фамилию и инициалы
```

```
2022-12-02 16:54:07.774181
```

```
Rogozhina N.A.
```

## Задания

### Задание 1.

1. Даны два списка одинаковых размеров из одинаковых элементов:

```
items = [1 5 6 0 9 8 7 2 3 4]
shuffled_items = [0 2 3 4 1 6 5 7 9 8]
```

2. Расставьте элементы (с помощью функции `sort()`) в списке `items` так, чтобы получился список `shuffled_items`

In [5]:

```
items = [1, 5, 6, 0, 9, 8, 7, 2, 3, 4]
shuffled_items = [0, 2, 3, 4, 1, 6, 5, 7, 9, 8]

print('Исходный список:', items)
items = sorted(items, key = lambda x: shuffled_items.index(x))
print('Получившийся список:', items)
print(items == shuffled_items)
```

```
Исходный список: [1, 5, 6, 0, 9, 8, 7, 2, 3, 4]
```

```
Получившийся список: [0, 2, 3, 4, 1, 6, 5, 7, 9, 8]
```

```
True
```

### Задание 2.

1. Создайте список `a`, состоящий из каких-то элементов.

2. Создайте список `b` такого же размера, как `a`, состоящий из каких-то элементов.
3. Выведите нумерованный список пар из элементов списков `a` и `b`.

In [13]:

```
a = [1,2,3,4,5,6,7]
b = [11,12,13,14,15,16,17]
print('Исходные списки:',a,b)
c = list(zip(a,b))
d = []
for i in range(len(a)):
    d.append([a[i], b[i]])
print('Решение через функцию:',c)
print('Для примера выведем 5-ую пару:', c[4])

print('Также можно сделать через цикл:', d)
```

Исходные списки: [1, 2, 3, 4, 5, 6, 7] [11, 12, 13, 14, 15, 16, 17]  
 Решение через функцию: [(1, 11), (2, 12), (3, 13), (4, 14), (5, 15), (6, 16), (7, 17)]  
 Для примера выведем 5-ую пару: (5, 15)  
 Также можно сделать через цикл: [[1, 11], [2, 12], [3, 13], [4, 14], [5, 15], [6, 16], [7, 17]]

### Задание 3.

1. Дан массив строк: ['agfkd.,f', 'Qksdf;sb&..', 'asdoo\*', 'bgf...d', 're54()kj[]].']
2. Создайте список, состоящий из количества точек в каждой строке. Выведите его
3. Создайте новый список, в котором будут **только строки, в которых более 2-х точек**. Выведите его

Циклами пользоваться нельзя.

In [1]:

```
arr = ['agfkd.,f', 'Qksdf;sb&..', 'asdoo*', 'bgf...d', 're54()kj[]].']
count = [dots.count('.') for dots in arr]
more_dots = [dot for dot in arr if dot.count('.') > 2]
print(count)
print(more_dots)
```

```
[1, 2, 0, 3, 1]
['bgf...d']
```

**Задание 4.** Напишите функцию, вычисляющую значения экспоненты по рекуррентной формуле  $e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ . Реализуйте контроль точности вычислений с помощью дополнительного параметра  $\varepsilon$  со значением по умолчанию (следует остановить вычисления, когда очередное приближение будет отличаться от предыдущего менее, чем на  $10^{-10}$ ).

In [3]:

```
from math import exp
```

In [7]:

```
from math import exp
from math import factorial

def EXPONENTA (x , eps = 10**( -10)):
    n = 1
    res = 1.0
    while True:
        tempres = res
        res += (x**n) / factorial(n)
        n += 1
        if (res - tempres) < eps:
            break
    return res

a = float(input('Введите показатель экспоненты: '))
print(exp(a))
print(EXPONENTA(a))
```

```
Введите показатель экспоненты: 1
2.718281828459045
2.71828182845823
```

In [6]:

```
# запустить после написания функции
assert round(EXPONENTA (a),9) == round(exp(a),9), "функция считает не корректно" # не удал
```