Задание 1.
Исходные данные:
Даны значения величины заработной платы заемщиков банка (zp) и значения их поведенческого кредитного скоринга (ks):
zp = [35, 45, 190, 200, 40, 70, 54, 150, 120, 110],
ks = [401, 574, 874, 919, 459, 739, 653, 902, 746, 832].
Найдите ковариацию этих двух величин с помощью элементарных действий, а затем с помощью функции cov из numpy
Полученные значения должны быть равны.
Найдите коэффициент корреляции Пирсона с помощью ковариации и среднеквадратичных отклонений двух признаков,
а затем с использованием функций из библиотек numpy и pandas.
Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> import pandas as pd
>>> zp = np.array([35, 45, 190, 200, 40, 70, 54, 150, 120, 110])
>>> ks = np.array([401, 574, 874, 919, 459, 739, 653, 902, 746, 832])
>>> zp_, ks_ = np.array(zp), np.array(ks)
>>> zp__, ks__ = pd.Series(zp_), pd.Series(ks_)
>>> cov_matrix = np.cov(zp_, ks_)
>>> print(cov_matrix)
[[ 3882.93333333 10175.37777778]
 [10175.37777778 33854.32222222]]
>>>
>>> zp_.var(ddof=1)
3882.9333333333334
>>>
>>> ks_.var(ddof=1)
33854.32222222223
>>>
>>> cov_zpks = zp__.cov(ks__)
>>> print(cov_zpks)
10175.377777777776
>>>
>>> cov_zpks = ks__.cov(zp__)
>>> print(cov_zpks)
10175.377777777776
>>>
>>> import scipy
>>> from scipy import stats
>>> r, p = scipy.stats.pearsonr(zp_, ks_)
>>> print(r)
0.8874900920739162
>>>
>>> print(p)
0.0006107546587257538
```

>>>

Задание 2.
Исходные данные:
Измерены значения IQ выборки студентов,
обучающихся в местных технических вузах:
131, 125, 115, 122, 131, 115, 107, 99, 125, 111.
Известно, что в генеральной совокупности IQ распределен нормально.
Найдите доверительный интервал для математического ожидания с надежностью 0.95.
Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([131, 125, 115, 122, 131, 115, 107, 99, 125, 111])
>>> a.mean()
118.1
>>> a.shape
(10,)
>>> a.std(ddof=1)
10.54566788359614
>>>
>>> a.std(ddof=1) / np.sqrt(a.shape[0])
3.3348329959851224
>>>
>>> def statistic(a: np.ndarray) -> float:
...     return (a.mean() - 1) / (0.95 / np.sqrt(a.shape[0]))
...
>>> alpha = 0.95
>>> import scipy
>>> from scipy import stats
>>> t1 = stats.norm.ppf(alpha / 2)
>>> t2 = stats.norm.ppf(1 - alpha / 2)
>>> print(t1, t2)
-0.06270677794321385 0.06270677794321385
```

```
>>>
>>> s = statistic(a)
>>> print(s)
389.79233053233395
>>>
```

Задание 3.

Исходные данные:

Известно, что рост футболистов в сборной распределен нормально
с дисперсией генеральной совокупности, равной 25 кв.см. Объем выборки равен 27,
среднее выборочное составляет 174.2. Найдите доверительный интервал для математического
ожидания с надежностью 0.95.

Решение:

```
Python 3.8.10 (default, Jun  2 2021, 10:49:15)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> a = np.array([172.25, 173.57, 174.2, 175.2, 173.59, 174.15, 174.85, 173.95, 171.95, 172.85,
173.85, 174.55, 173.25, 174.92, 175.25, 174.35, 173.68, 174.49, 174.25, 173.78, 173.55, 175.15,
175.85, 175.97, 174.95, 174.80, 174.2])
>>> a.mean()
174.2
>>>
>>> a.std(ddof=1)
0.9681703600884745
>>>
>>> a.shape
(27,)
>>>
>>> a.std(ddof=1) / np.sqrt(a.shape[0])
0.18632447267283253
>>>
>>> def statistic(a: np.ndarray) -> float:
...     return (a.mean() - 1) / (0.95 / np.sqrt(a.shape[0]))
...
>>> alpha = 0.95
>>> import scipy
>>> from scipy import stats
```

```
>>> t1 = stats.norm.ppf(alpha / 2)
>>> t2 = stats.norm.ppf(1 - alpha / 2)
>>> print(t1, t2)
-0.06270677794321385 0.06270677794321385
>>>
>>> s = statistic(a)
>>> print(s)
947.3406311713566
>>>
>>>
```