**Задание1**

Объясните результат операции:

arr=[1,2,3,4,5,6,7,8,9,10]

print(arr[:5:-2])

#Из последнего элемента -Отсчитайте четвертую в обратном порядке.#

#шага - 2#

**Задание2**

Получите перевернутый список

data = list(range(10))

print(data[::-1])

**Задание3**

Получите массив без первого и двух последних элементов:

data = list(range(10))

print(data[1:8:1])

**Задание4**

Получите массив без крайних рядов и колонок (по одной с каждой стороны)

data = np.random.randint(low=0, high=10, size=(5, 6))

print(data)

**Задание5**

Произведите перемешивание колонок с помощью функции numpy.random.permutation():

data = np.random.randint(low=0, high=10, size=(3, 6))

print(data)

a=np.random.permutation(data)

print(a)

**Задание6**

# Определите, имеется ли хотя бы одно значение больше пяти и меньше восьми в массиве:

import numpy as np

arr = np.array([[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]])

print(arr2.any())

**Задание7**

# Напишите операцию получения суммы элементов по всем стобцам:#

data = np.random.randint(low=0, high=5, size=(3, 6))

print(np.sum(data))

**Задание8**

# Объедините три массива вдоль колонок#

x = np.arange(9).reshape((3, 3))

y = np.arange(12).reshape((3, 4))

z = np.arange(15).reshape((3, 5))

print(np.concatenate((x, y,z), axis=1))

**Задание9**

# Напишите реализацию добавления стобца единиц к матрице слева:#

x = np.arange(1, 10).reshape((3, 3))

new\_x = np.ones((3, 1))

b = np.hstack([new\_x, x])

print(b)

**Задание10**

По такому же принципу именования найдите функцию, которая возвращает индекс максимального/минимального (в завимости от варианта) элемента в массиве.

Воспользуйтесь найденной функцией и найдите индексы максимальных элементов в каждом столбце 2D матрицы:

arr = np.array([

[11, -1, 1, 3, 5],

[6, 1, 0, -1, 10],

[1, 3, 2, 3, 6],

[5, 2, 7, 8, 9],

])

print(np.argmax(arr, axis=0))

print(np.argmin(arr, axis=0))

**# Задание 11**

Напишите код сортировки массива по столбцам:

arr = np.array([

[11, 2, 1, 3, 5],

[6, 2, 0, -1, 10],

[1, 3, 2, 3, 5],

[5, 2, 7, 8, 9],

])

print( np.argsort(x, axis=0))

**# Задачки1**

Создайте массив, состоящий из случайных элементов в диапазоне $[-10; 20]$ размером $(5, 7)$:

np.random.randint(-10, 10, size=(5, 7))

**# Задачки2**

**# Добавьте границу в виде нулей со всех сторон 2D массива:#**

data = np.arange(1, 16).reshape((3, 5))

data = np.pad(data, pad\_width=1, mode='constant', constant\_values=0)

print(data)

**# Задачки3**

**# Разберитесь с функцией np.clip() и произведите ограничение массива, чтобы в нем значения были не более шести и не менее двух:**

data = np.arange(1, 16).reshape((3, 5))

data = np.clip(data, 2, 6)

print(data)

**# Задачки4**

# Произведите модификацию массива так, чтобы значение в колонке результата соответствовало исходному значению, возведенному в степень индекса колонки:

data = np.full((3, 5), fill\_value=2, dtype=np.int32)

vec = np.arange(5)

for i\_row in range(data.shape[0]):

result[i\_row,:] = pow(data[i\_row,:] ,vec)

print(result)

**# Задачки5**

**# Умножьте каждую колонку на ее индекс:**

data = np.full((3, 5), fill\_value=2, dtype=np.int32)

vec = np.arange(5)

for i\_row in range(data.shape[0]):

result[i\_row,:] = data[i\_row,:]\*vec

print(result)

# Задачки6

# Умножьте каждый четный ряд в матрице на вектор:

**import numpy as np**

**data = np.repeat(np.arange(1, 8), 4).reshape(7, 4)**

**vec = np.array([1, 2, 1, 2])**

**for i in range(data.shape[0]):**

**if i%2==0:**

**data[i] = data[i]\*vec**

**print(data)**

**# Задачки7**

**# Определите индексы двух наибольших элементов в матрице по каждому ряду:**

data = np.array([[20, 22, 2, 14, 25, 8],

[ 7, 25, 23, 3, 22, 15],

[ 8, 16, 9, 22, 0, 9],

[ 4, 24, 24, 28, 3, 40]])

idx = (-data).argsort()[:4]

print(idx[:,[0,1]])

**# Задачки8**

**# Выберите две случайные колонки из массива:**

data = np.array([[20, 22, 2, 14, 25, 8],

[ 7, 25, 23, 3, 22, 15],

[ 8, 16, 9, 22, 0, 9],

[ 4, 24, 24, 28, 3, 40]])

data[np.random.randint(0,data.shape[0],2)]

**# Задачки9**

**# Создайте массив размером $(10, 3)$, который состоит из повторяющихся рядов исходного массива:**

data = np.arange(9).reshape(3, 3)

data = np.tile(data,(3,1))

print(data)

**# Задачки10**

**# Соберите массив, состоящий из элементов 2й и 4й колонок и строк с 1й по 3ю:**

data = np.array([[20, 22, 2, 14, 25, 8],

[ 7, 25, 23, 3, 22, 15],

[ 8, 16, 9, 22, 0, 9],

[ 14, 9, 33, 21, 7, 6],

[ 4, 24, 24, 28, 3, 40]])

print([data[1,2],data[1,4]])

print([data[2,2],data[2,4]])

print([data[3,2],data[3,4]])

**# Задачки11**

**# Инвертируйте порядок элементов в последних двух колонках:**

import numpy as np

data = np.array([[20, 22, 2, 14, 25, 8],

[ 7, 25, 23, 3, 22, 15],

[ 8, 16, 9, 22, 0, 9],

[ 14, 9, 33, 21, 7, 6],

[ 4, 24, 24, 28, 3, 40]])

b = data[:, [-2, -1]]

you = np.flipud(b)

zuo = np.delete(data, [4,5], axis=1)

print(np.hstack((zuo,you)))