**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

###### ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

###### КЕМЕРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

**ИНСТИТУТ ЦИФРЫ**

**ОТЧЁТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ № 1**

Студентки 3 курса группы ФИТ-211

**Колесник Полины Олеговны**

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Руководитель:

Ассистент Подберезен Г. И.

Работа защищена

« »

“ ” 2023 г.

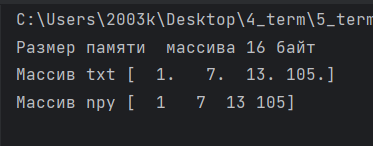
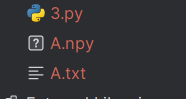
Кемерово 2023 г.

**ОТЧЁТ О ПРОДЕЛАННОЙ РАБОТЕ**

**1 задание**

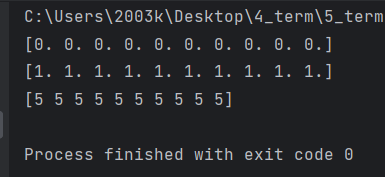
1. Массив со значениями 1, 7, 13, 105. Посчитайте и выведите на экран размер памяти, который он занимает. Сохраните массив в текстовый и бинарный файлы. Затем загрузите его и выведите на экран.

import numpy as np  
  
A = np.array([1, 7, 13, 105])  
  
memory\_size = A.nbytes  
print("Размер памяти массива", memory\_size, "байт")  
  
np.savetxt('A.txt', A)  
  
np.save("A.npy", A)  
  
loaded\_txt = np.loadtxt("A.txt")  
loaded\_npy = np.load("A.npy")  
  
print('Массив txt', loaded\_txt)  
print('Массив npy', loaded\_npy)

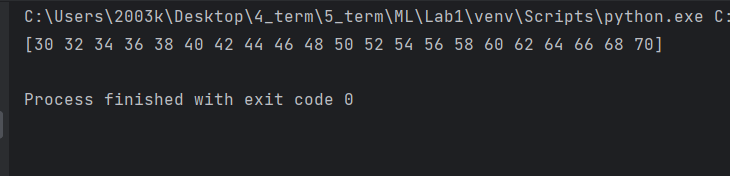
1. Три массива: из 10 нулей, 10 единиц, 10 пятерок.

import numpy as np  
  
Arr\_zero = np.zeros(10)  
Arr\_one = np.ones(10)  
Arr\_five = np.full(10, 5)  
  
print(Arr\_zero)  
print(Arr\_one)  
print(Arr\_five)



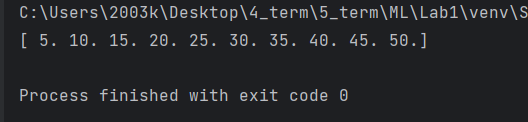
1. Массив последовательности четных целых чисел от 30 до 70.

import numpy as np  
  
Arr\_even = np.arange(30, 71, 2)  
  
print(Arr\_even)



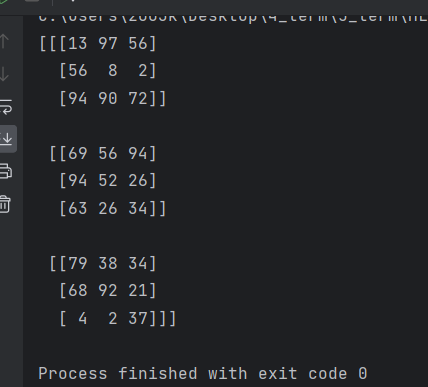
1. Массив из 10 элементов со значениями, равномерно распределенными между 5 и 50.

import numpy as np  
  
Arr\_ravno = np.linspace(5, 50, 10)  
  
print(Arr\_ravno)



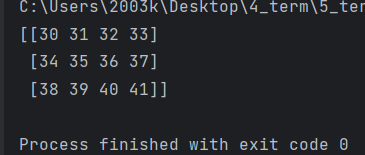
1. Массив 3x3x3 из 27 случайных чисел в диапазоне от 1 до 100.

import numpy as np  
  
Arr\_rand = np.random.randint(1, 101, (3, 3, 3))  
  
print(Arr\_rand)



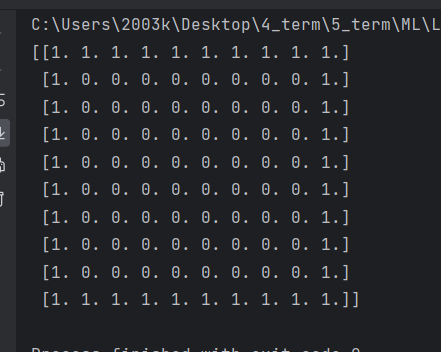
1. Массив 3х4, заполненной значениями от 30 до 41.

import numpy as np  
  
Arr\_range = np.arange(30, 42).reshape(3, 4)  
  
print(Arr\_range)



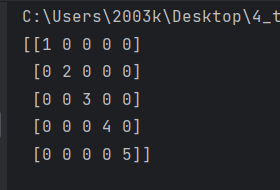
1. Массив 10x10, в котором элементы на границах равны 1, а внутри 0.

import numpy as np  
  
Arr\_square = np.zeros((10, 10))  
  
Arr\_square[[0, -1], :] = 1  
Arr\_square[:, [0, -1]] = 1  
  
print(Arr\_square)



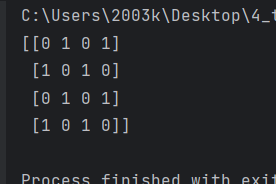
1. Массив 5x5 из нулей с элементами на главной диагонали равными 1, 2, 3, 4, 5.

import numpy as np  
  
Arr\_diag = np.diag(np.arange(1, 6), k=0)  
  
print(Arr\_diag)



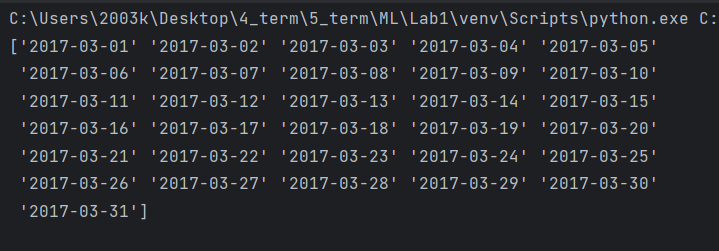
1. Массив 4x4, в котором 0 и 1 расположены в шахматном порядке, с нулями на главной диагонали.

import numpy as np  
  
arr\_1 = np.ones((4, 4), np.int8)  
arr\_1[4 % 2::2, ::2] = 0  
arr\_1[(4 + 1) % 2::2, 1::2] = 0  
  
print(arr\_1)



1. Массив, заполненный всеми днями марта 2017 года. Тип данных массива – дата.

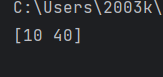
import numpy as np  
  
dates\_range = np.arange('2017-03-01', '2017-04-01', dtype='datetime64[D]')  
  
print(dates\_range)



**2 задание**

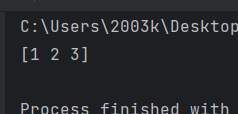
1. Дано два массива. Получите массив, состоящий из их общих элементов. Например, для массивов [0 10 20 40 60] и [10 30 40] ответом будет [10, 40].

import numpy as np  
  
A = np.array([0, 10, 20, 40, 60])  
B = np.array([10, 30, 40])  
  
print(np.intersect1d(A, B))



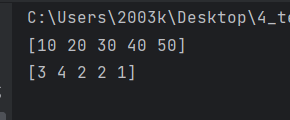
2. Дан массив. Получите массив его уникальных элементов.  
Исходный массив: [10 10 20 20 30 30]  
Массив его уникальных элементов: [10 20 30]   
Исходный массив:   
[[1 1]   
[2 3]]   
Массив его уникальных элементов: [1 2 3]

import numpy as np  
  
A = np.array([[1, 1], [2, 3]])  
  
print(np.unique(A))



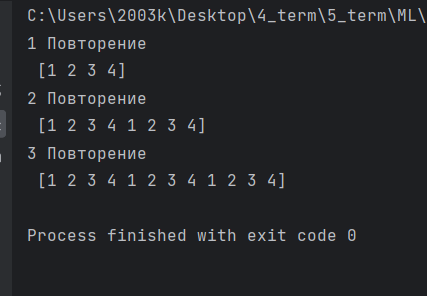
3. Дан массив. Получите два массива: его уникальные элементы и их частоты.   
Исходный массив: [10 10 20 10 20 20 20 30 30 50 40 40]   
Два массива: [10 20 30 40 50], [3 4 2 2 1]

import numpy as np  
  
A = np.array([10, 10, 20, 10, 20, 20, 20, 30, 30, 50, 40, 40])  
  
unique, frequencies = np.unique(A, return\_counts=True)  
  
print(unique)  
print(frequencies)



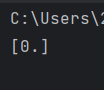
4. Дан массив. Получите массив путем повторения данного.   
Образец массива: [1, 2, 3, 4]   
Ожидаемый результат:   
1 повторение   
[1, 2, 3, 4]   
2 повторения   
[1, 2, 3, 4, 1, 2, 3, 4]   
3 повторения   
[1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4]

import numpy as np  
  
A = np.array([1, 2, 3, 4])  
  
for i in range(1, 4):  
 print(i, "Повторение\n", np.tile(A, i))



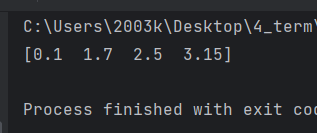
5. Дан массив, в котором имеются значения nan. Получите массив, удалив все значения nan.  
Исходный массив: [200. 300. nan nan nan 700.]   
После удаления значений nan: [200. 300. 700.]   
Исходный массив:   
[[1. 2. 3.]   
[nan 0. nan]   
[6. 7. nan]]   
После удаления значений nan: [1. 2. 3. 0. 6. 7.]

from cmath import nan  
  
import numpy as np  
  
A = np.array([nan, 0, nan])  
  
print(A[~np.isnan(A)])



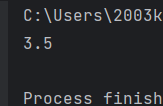
6. Дан массив. Получите массив его k наименьших значений.   
Оригинальный массив: [1. 7. 8. 2. 0.1 3. 15. 2.5]  
Его 4 наименьших значения: [0.1 1. 2. 2.5]

import numpy as np  
  
  
def sorted\_arr(k):  
 A = np.array([1.7, 8.2, 0.1, 3.15, 2.5])  
  
 print(np.sort(A)[:k])  
  
  
sorted\_arr(4)



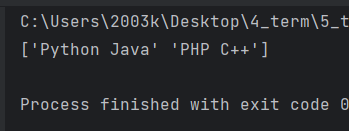
7. Дан массив и некоторое число. Найдите ближайший по значению к числу элемент массива.  
Исходный массив: [0.5 1.8 2.1 3.5 4.87 5.13 6.49]   
Значение для сравнения: 3,09066280756759   
Ответ: 3.5

import numpy as np  
  
A = np.array([0.5, 1.8, 2.1, 3.5, 4.87, 5.13, 6.49])  
numbers = 3.09066280756759  
  
print(A[np.abs(A - numbers).argmin()])



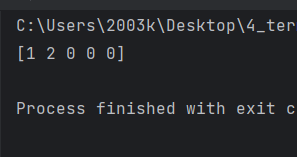
8. Дано два массива со строками. Получите массив при помощи поэлементного объединения.   
1 массив: ['Python' 'PHP']   
2 массив: ['Java' 'C ++']   
Результат: ['Python Java' 'PHP C ++']

import numpy as np  
  
A = np.array(['Python', 'PHP'])  
B = np.array([' Java', ' C++'])  
  
print(np.char.add(A, B))



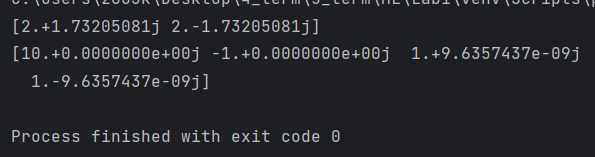
9. Дан массив строк. Получите массив с частотой встречаемости буквы «P».   
Оригинальный массив: ['Python' 'PHP' 'JS' ' examples' 'html']   
Количество 'P': [1 2 0 0 0]

import numpy as np  
  
B = np.array(['Python', 'PHP', 'JS', ' examples', 'html'])  
  
print(np.char.count(B, 'P'))



10. Найдите корни следующих полиномов:   
a. x^2 - 4х + 7   
b. x^4 - 11х^3 + 9х^2 + 11х - 10

import numpy as np  
  
# a) pow(x, 2) - 4 \* x + 7  
  
B = np.array([1, -4, 7])  
  
print(np.roots(B))  
  
# b) pow(x, 4) - 11\* pow(x, 3) + 9\*pow(x, 2) + 11\*x -10  
  
A = np.array([1, -11, 9, 11, -10])  
  
print(np.roots(A))



**3 задание**

Напишите программу, которая будет решать систему линейных уравнений вида:

a11\*x +a12\*y = b1 (1)

a21\*x+a22\*y = b2 (2)

Коэффициенты и правую часть вводит пользователь. Программа выводит решение системы. Для нахождения решения используйте numpy.linalg.solve.

Система не имеет решения (или имеет их бесконечно много), если определитель матрицы коэффициентов равен 0. В этом случае необходимо выводить соответствующее сообщение. Для нахождения определителя используйте функцию numpy.linalg.det.

import numpy as np  
  
a11, a12, a21, a22, b1, b2 = map(int, input().split())  
  
a = np.array([[a11, a12], [a21, a22]])  
b = np.array([b1, b2])  
  
if np.linalg.det(a) == 0:  
 print('Нет решения')  
else:  
 print(np.linalg.solve(a, b))

