 МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Комп’ютерний практикум №2**

з дисципліни «Методи та технології обчислювального інтелекту»

на тему: «Основи роботи з бібліотекою Numerical Python»

**Виконала:**

студентка гр. БС-01мп

Харченко Наталя

**Перевірив:**

доц. Федорін І.В.

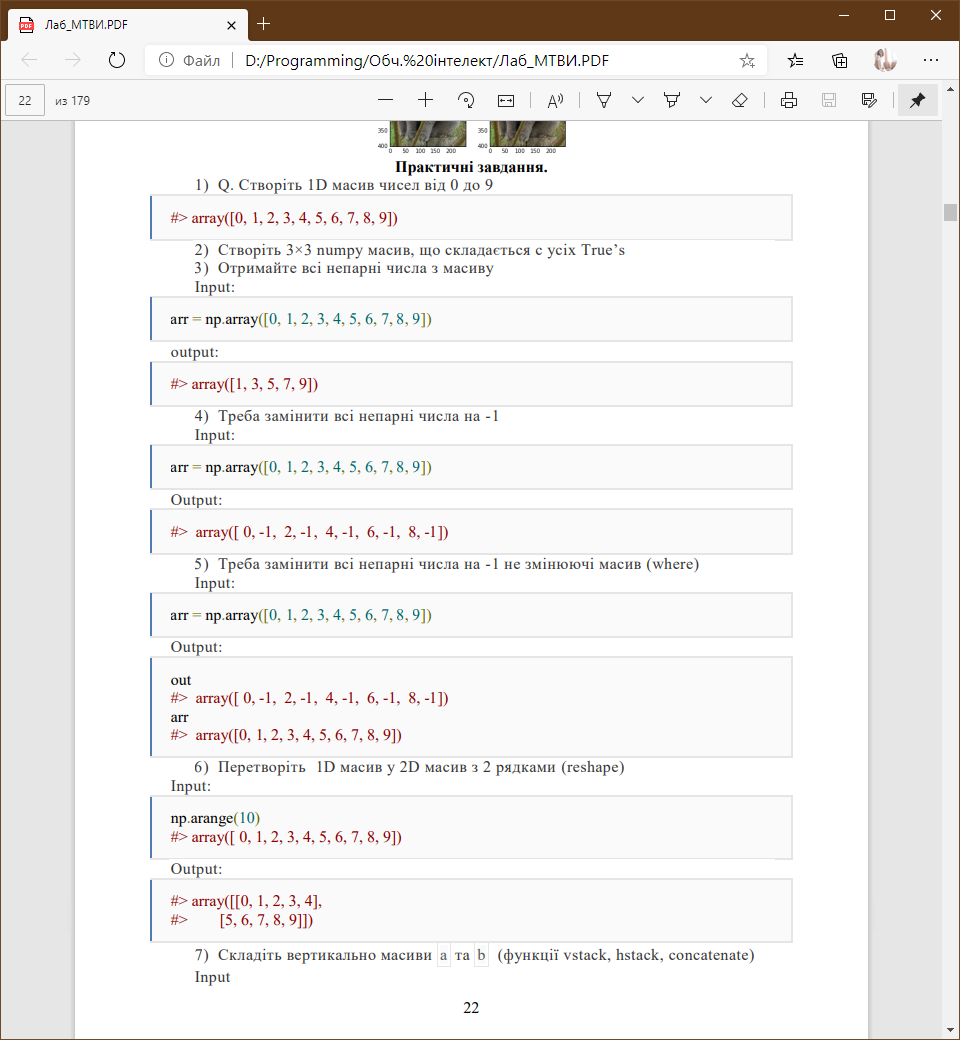
Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2020

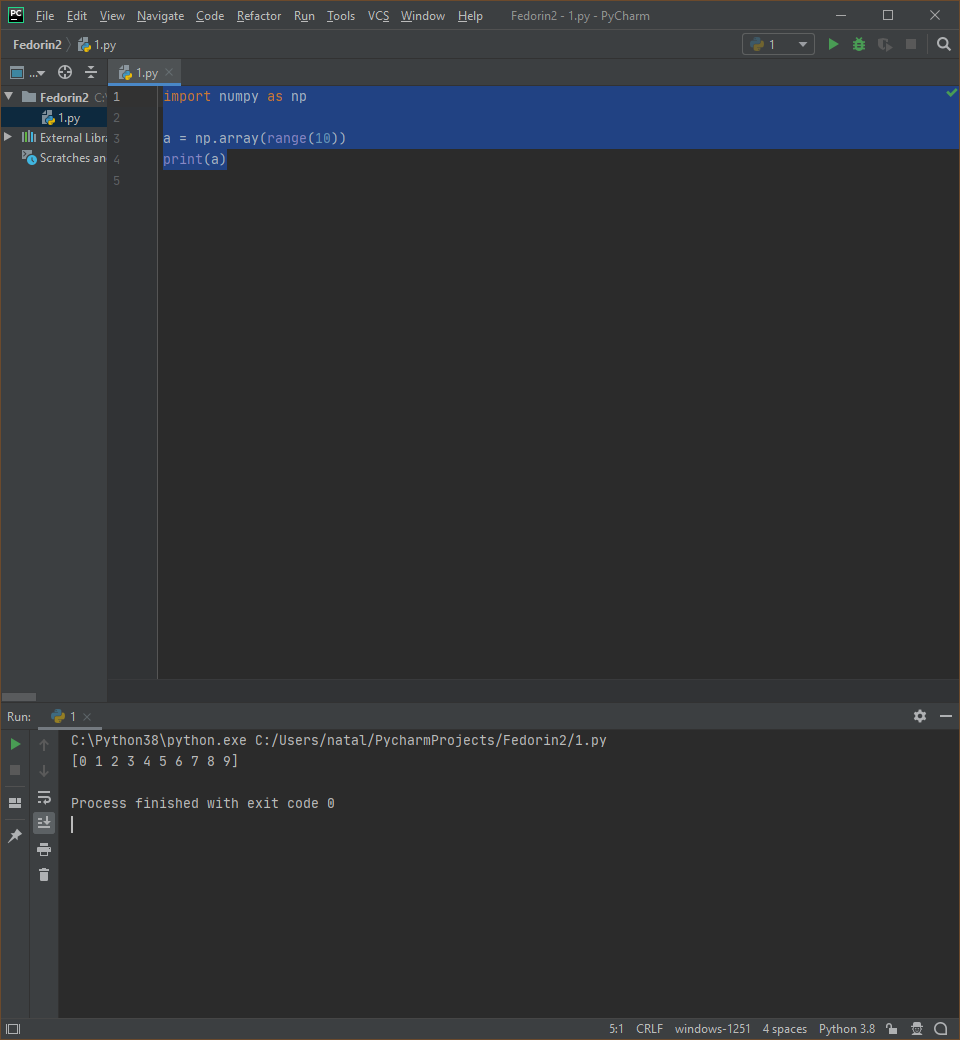
**Практичні завдання**

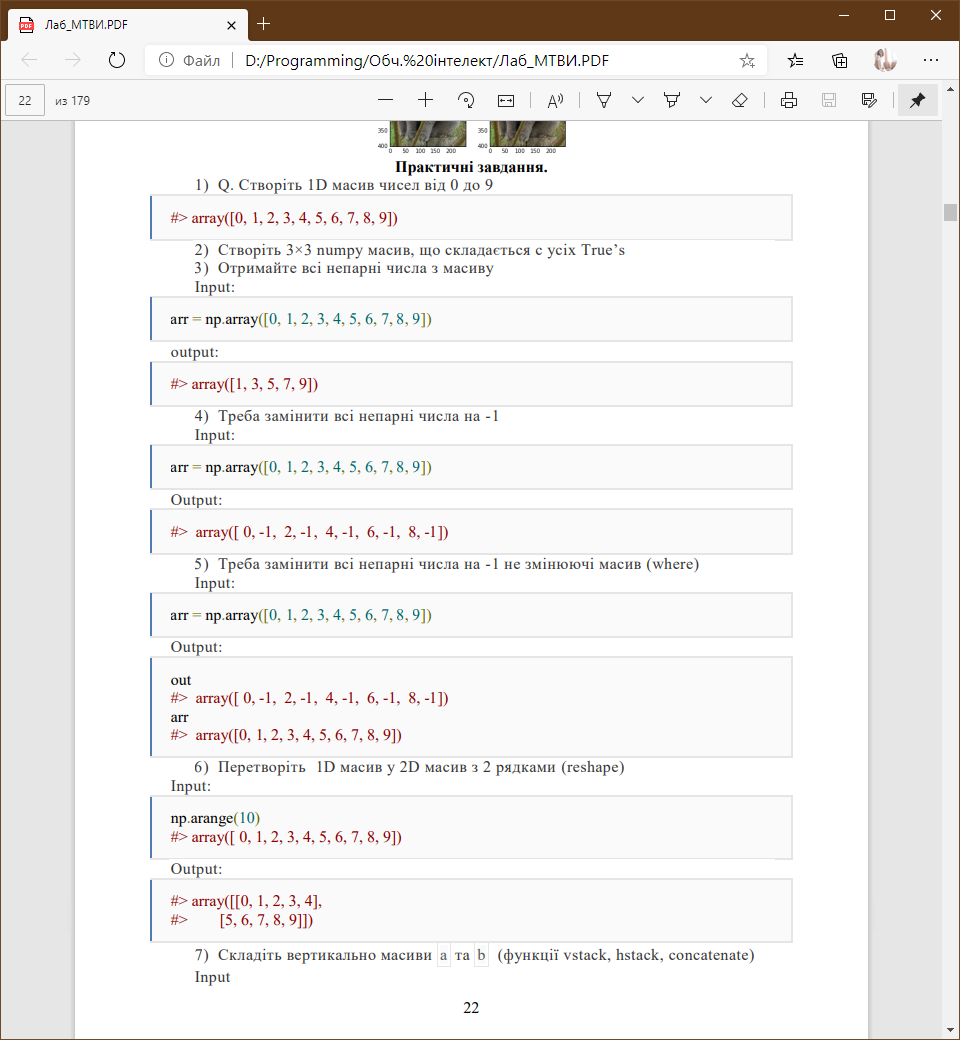


**Програмна реалізація:**

import numpy as np  
  
a = np.array(range(10))  
print(a)

**Результат:**

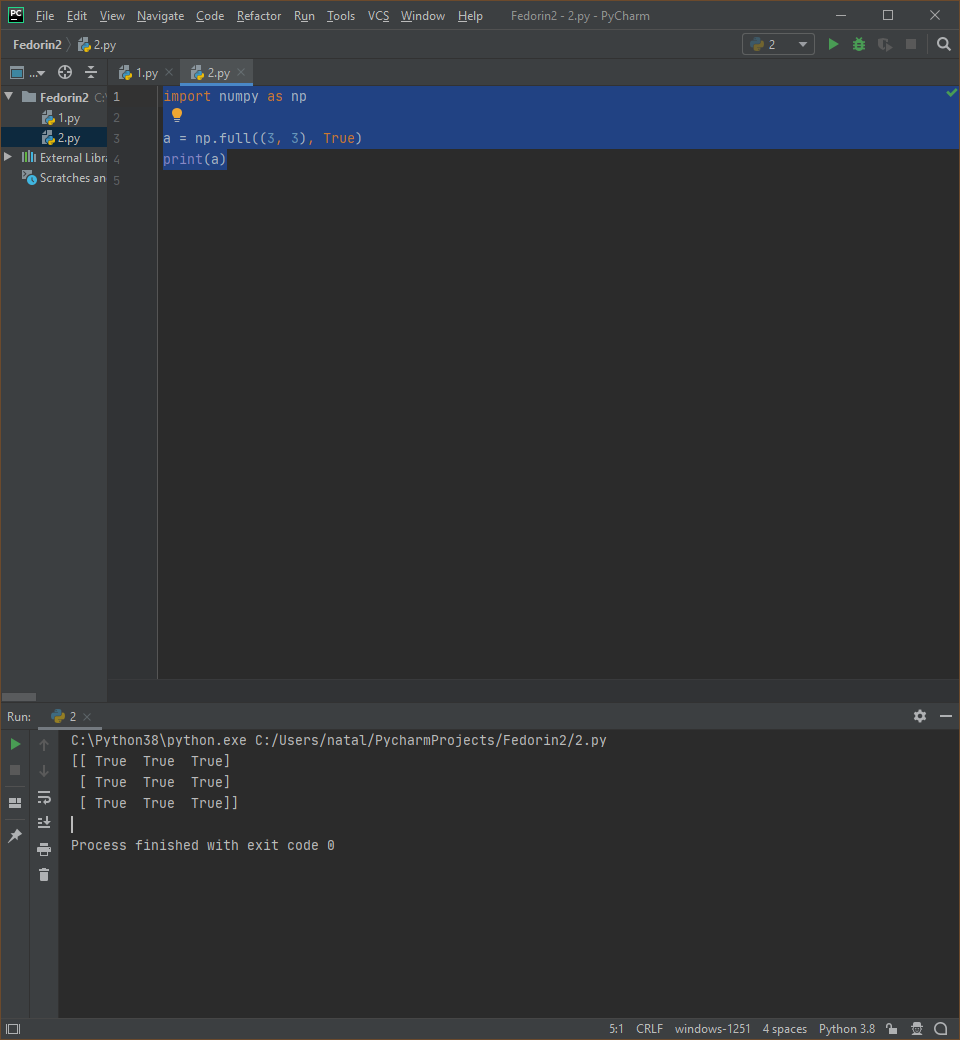


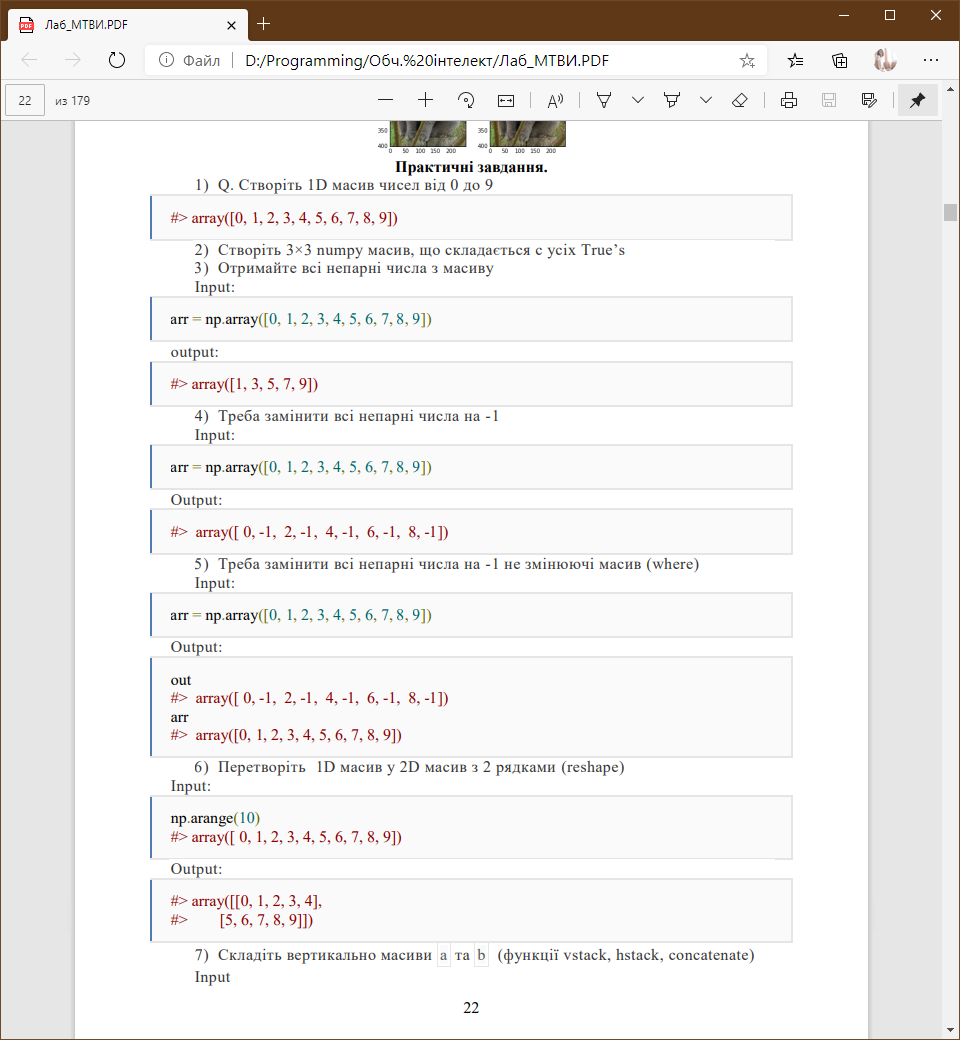


**Програмна реалізація:**

import numpy as np  
  
a = np.full((3, 3), True)  
print(a)

**Результат:**

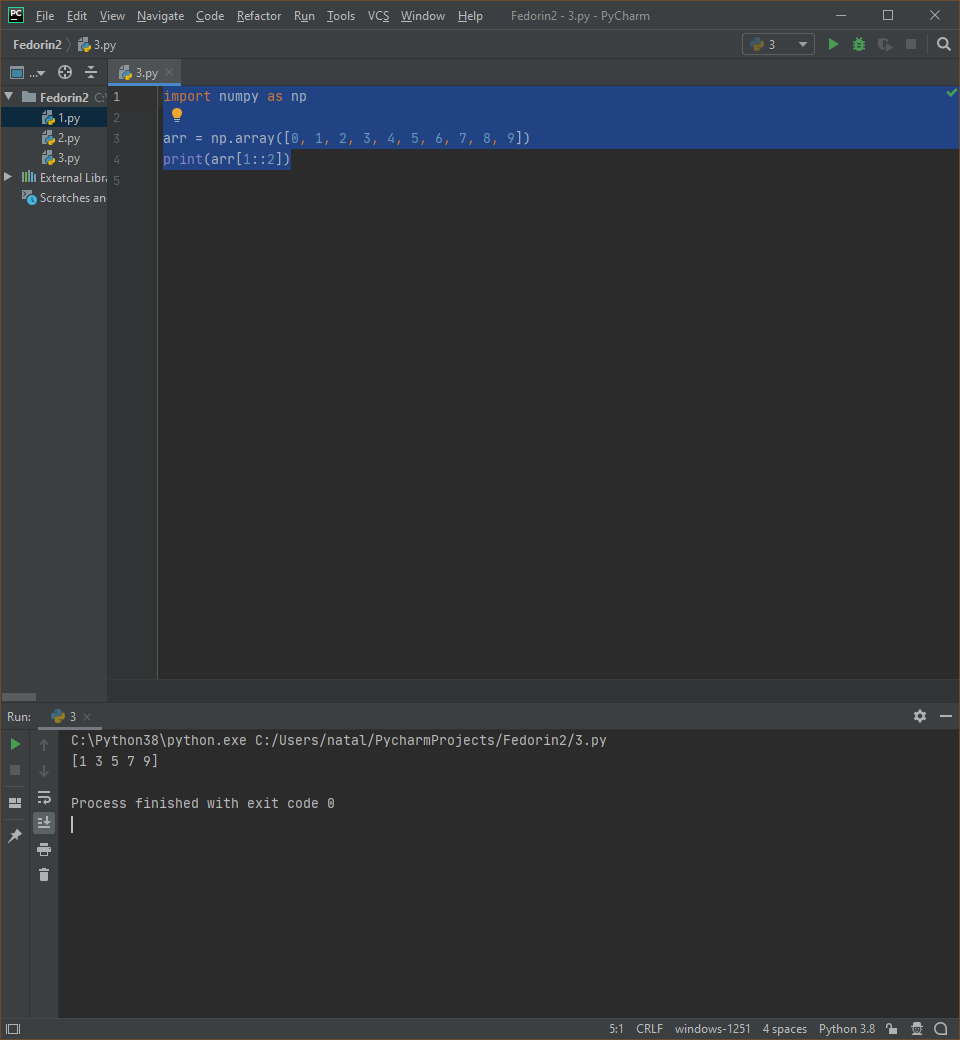


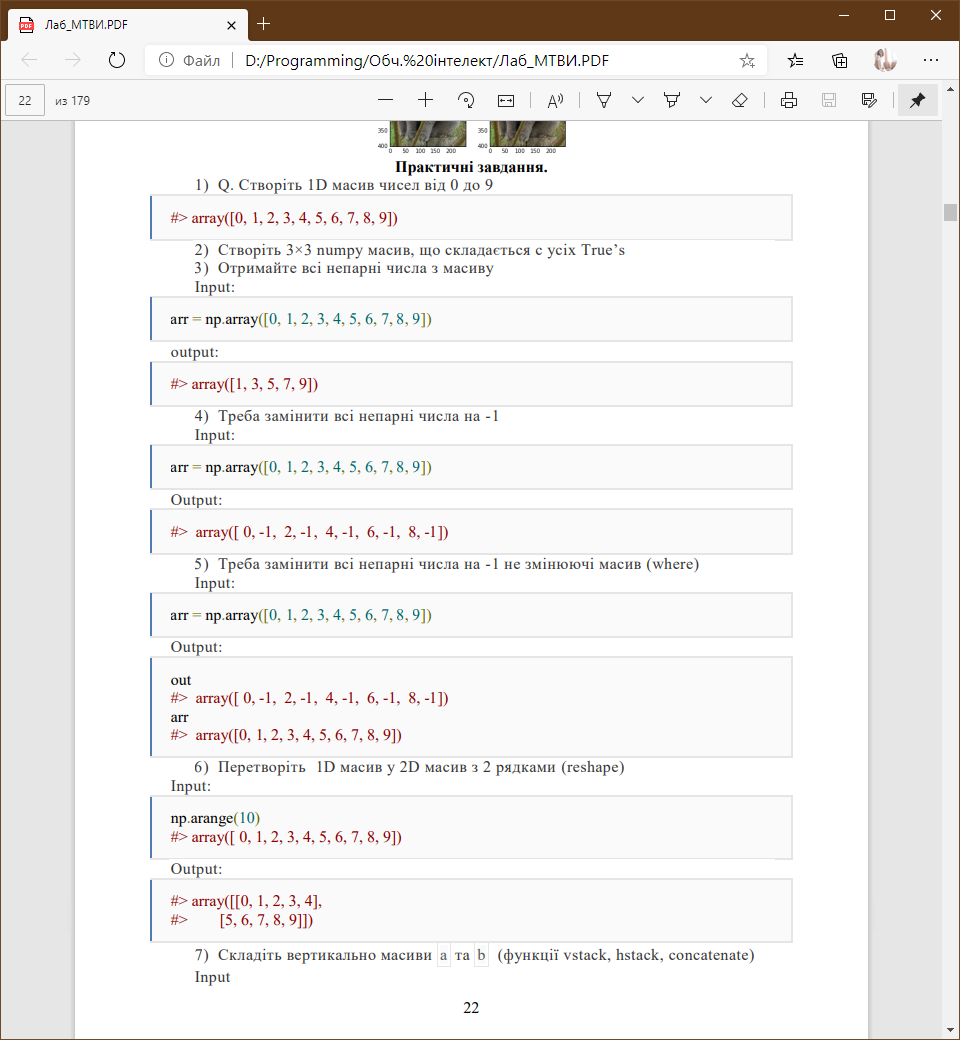


**Програмна реалізація:**

import numpy as np  
  
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])  
print(arr[1::2])

**Результат:**

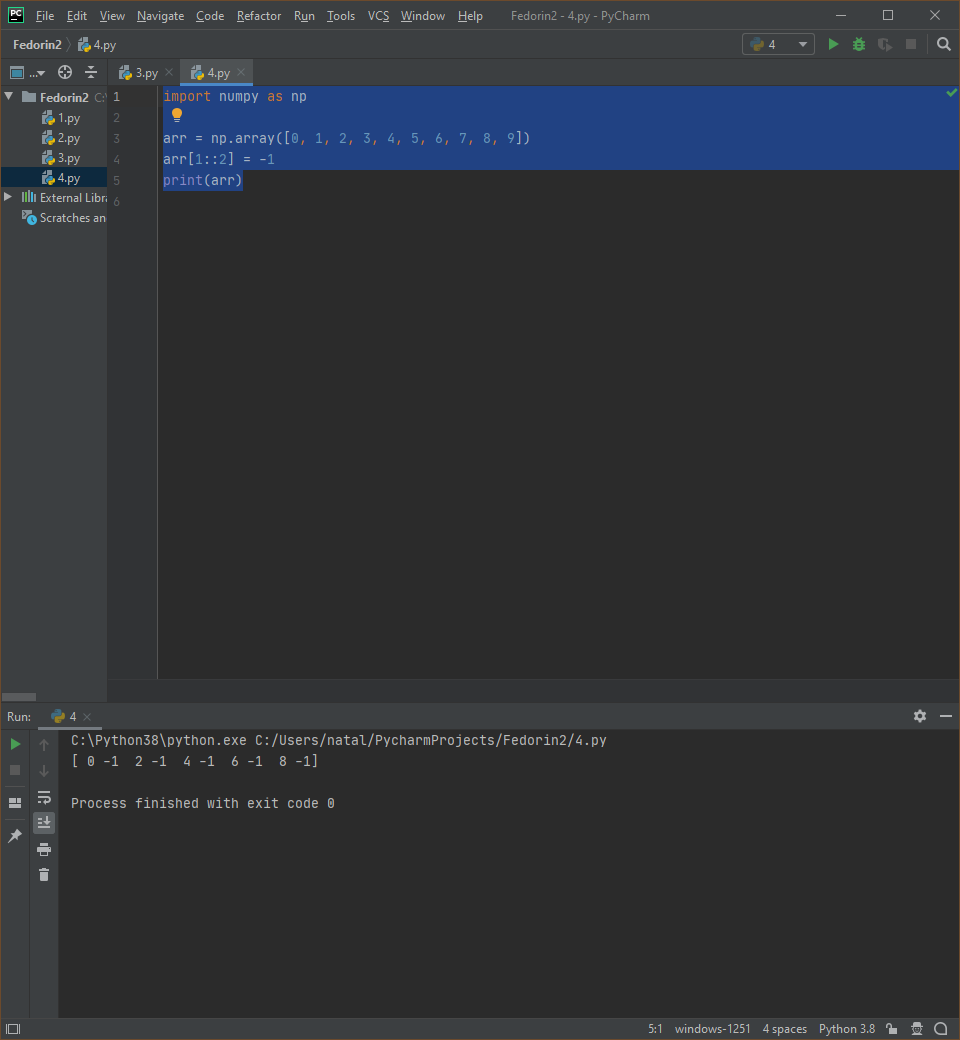


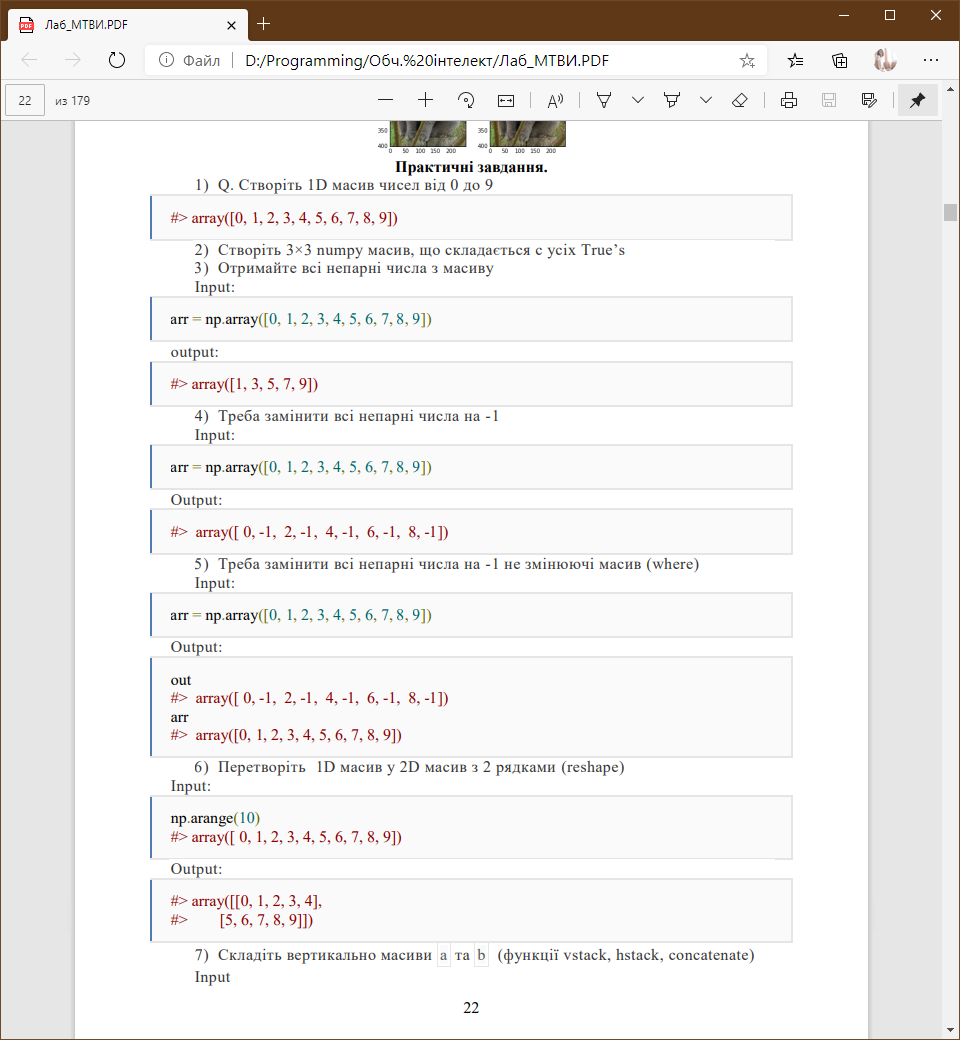


**Програмна реалізація:**

import numpy as np  
  
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])  
arr[1::2] = -1  
print(arr)

**Результат:**

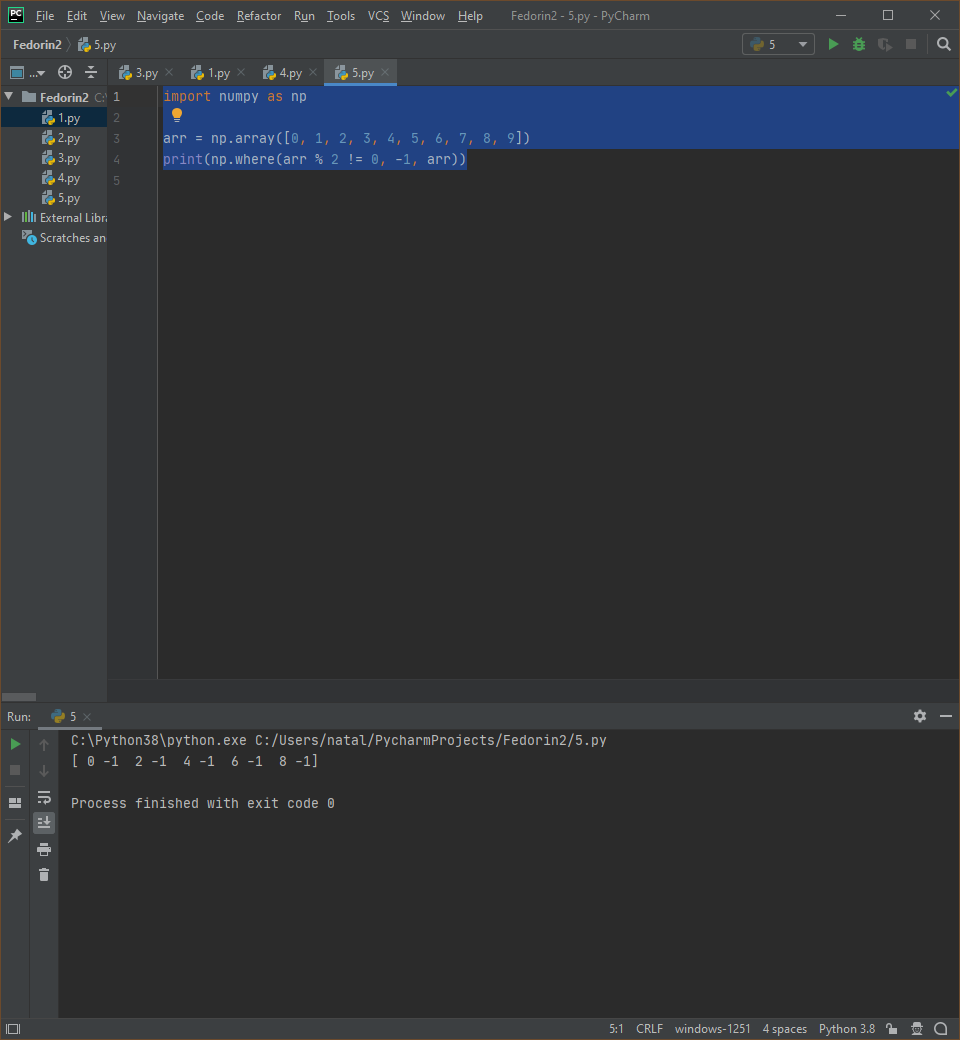


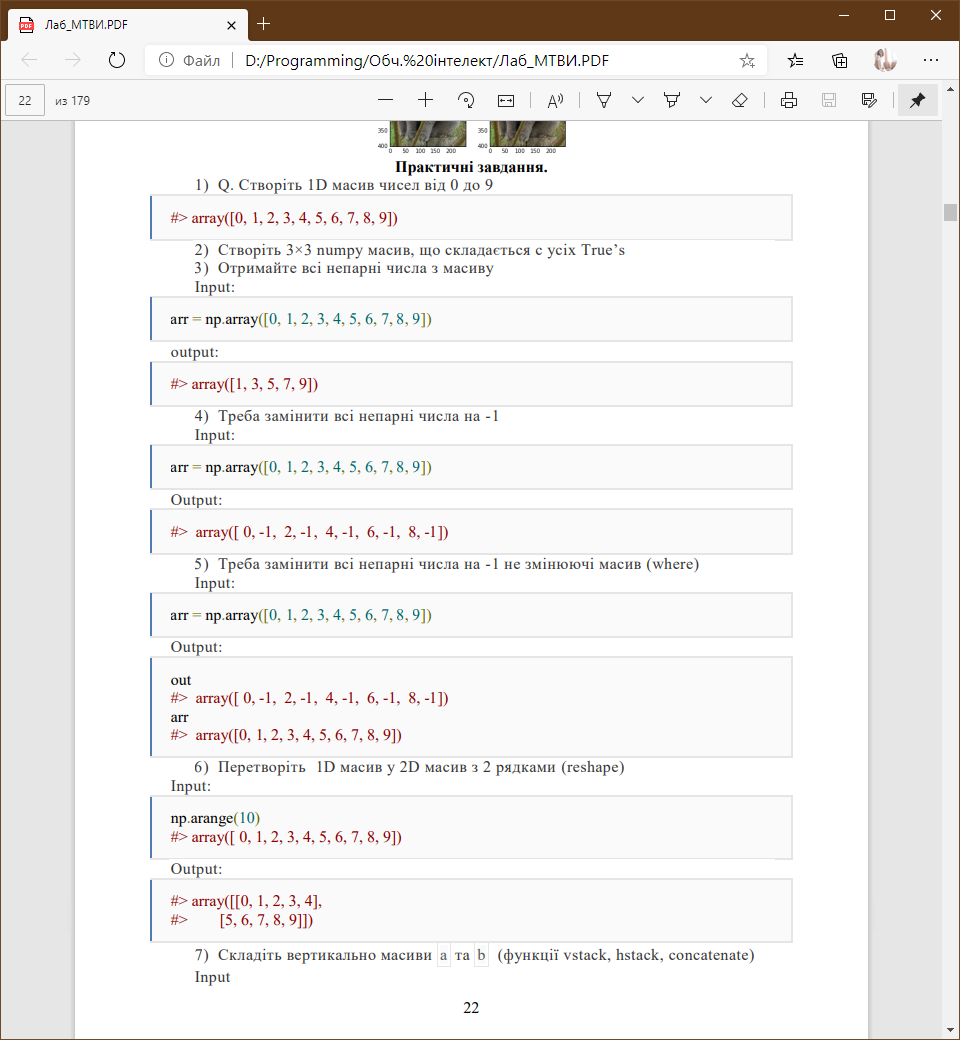


**Програмна реалізація:**

import numpy as np  
  
arr = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9])  
print(np.where(arr % 2 != 0, -1, arr))

**Результат:**

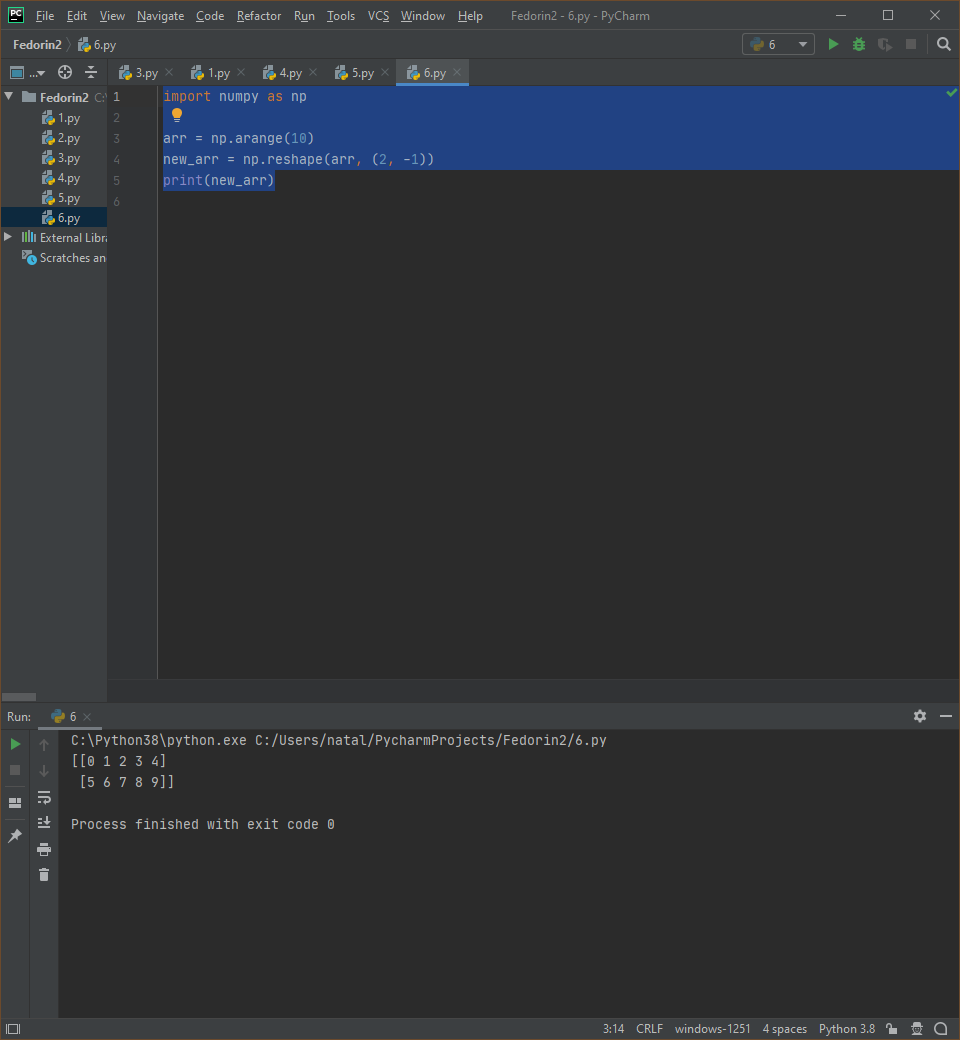


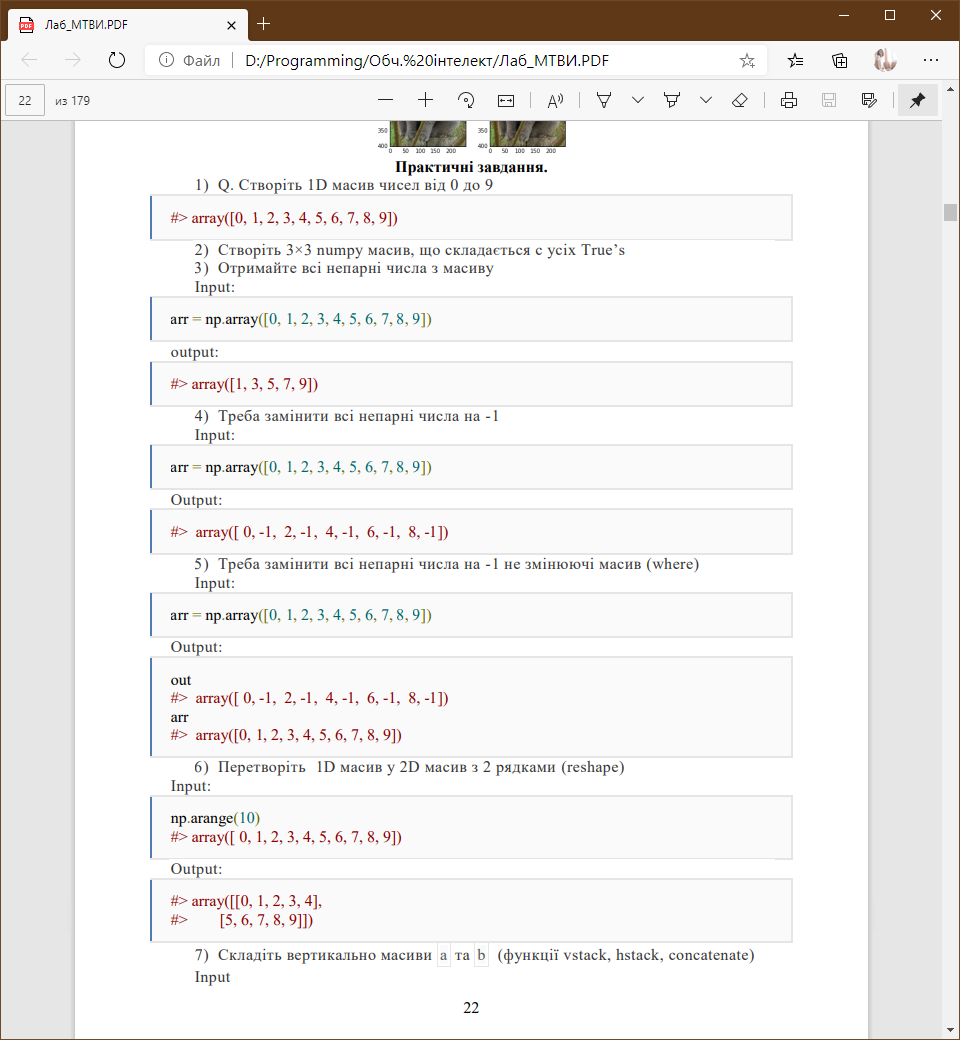


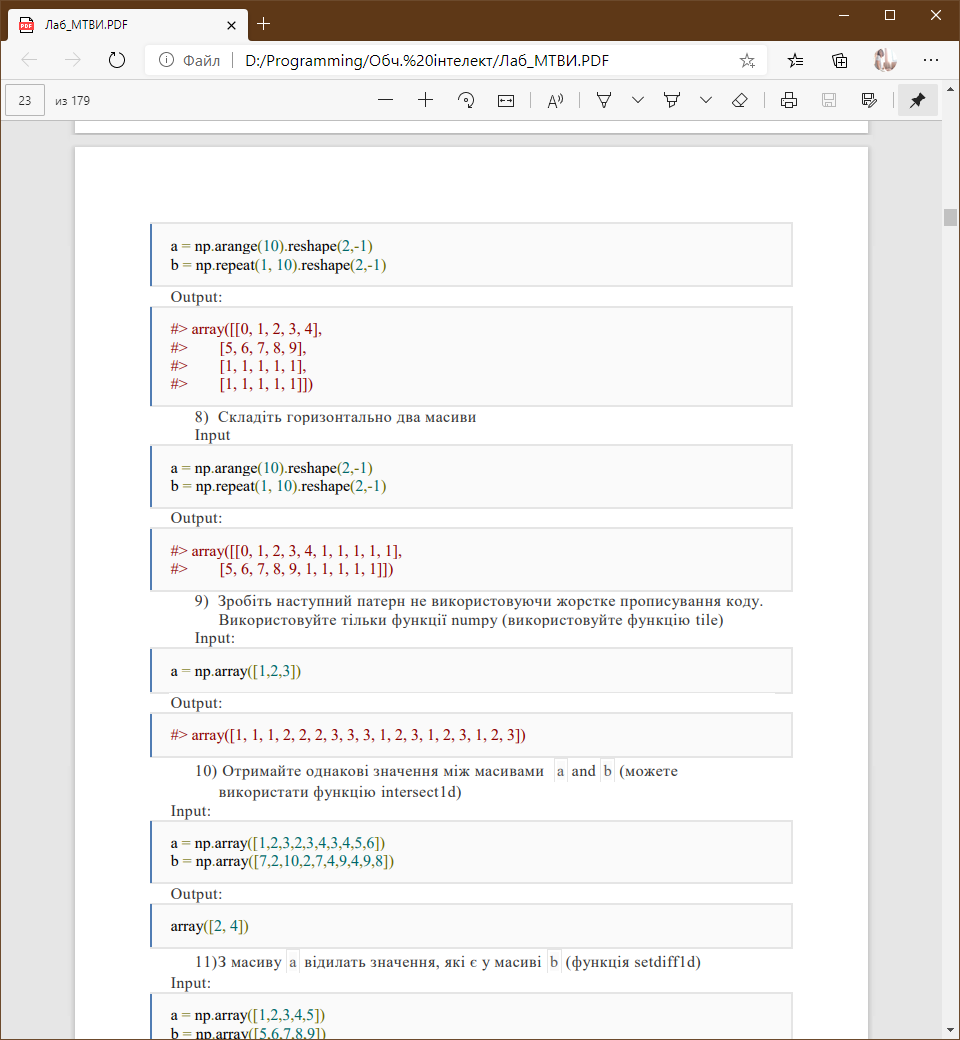
**Програмна реалізація:**

import numpy as np  
  
arr = np.arange(10)  
new\_arr = np.reshape(arr, (2, -1))  
print(new\_arr)

**Результат:**



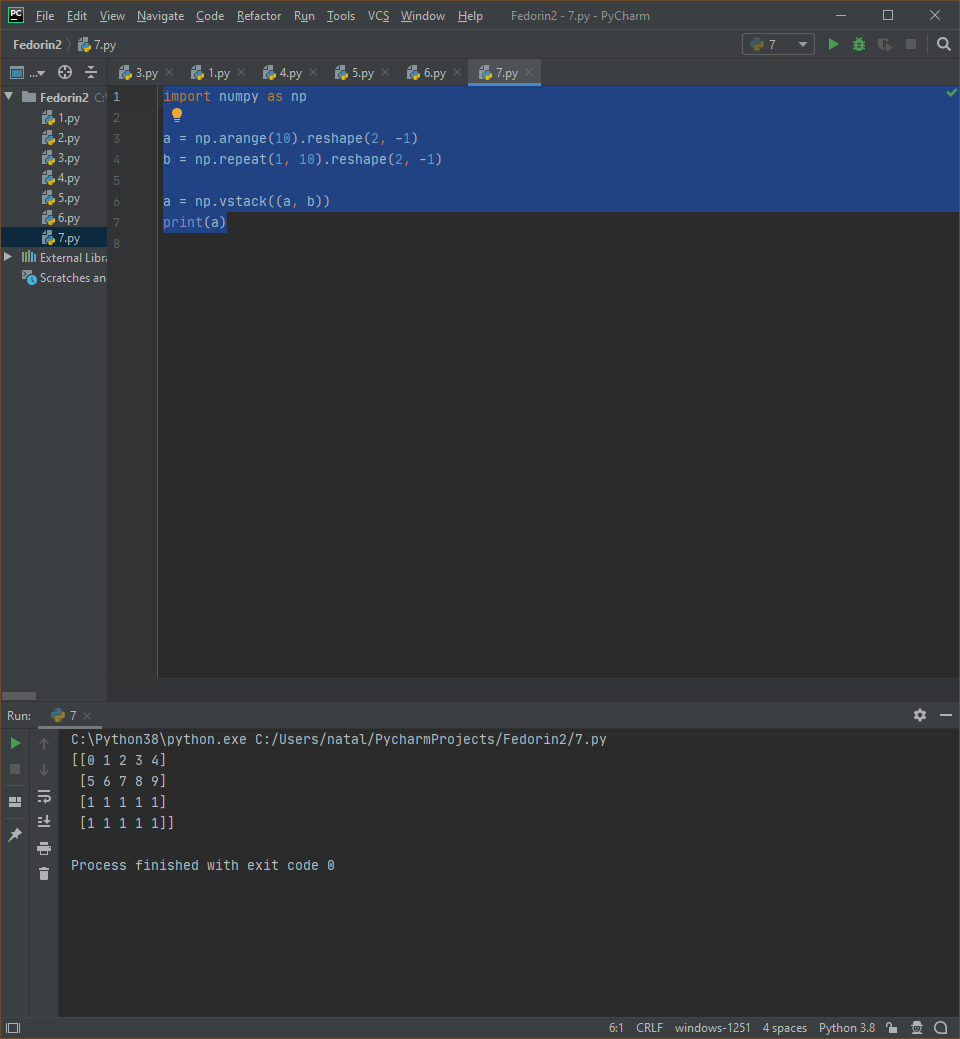


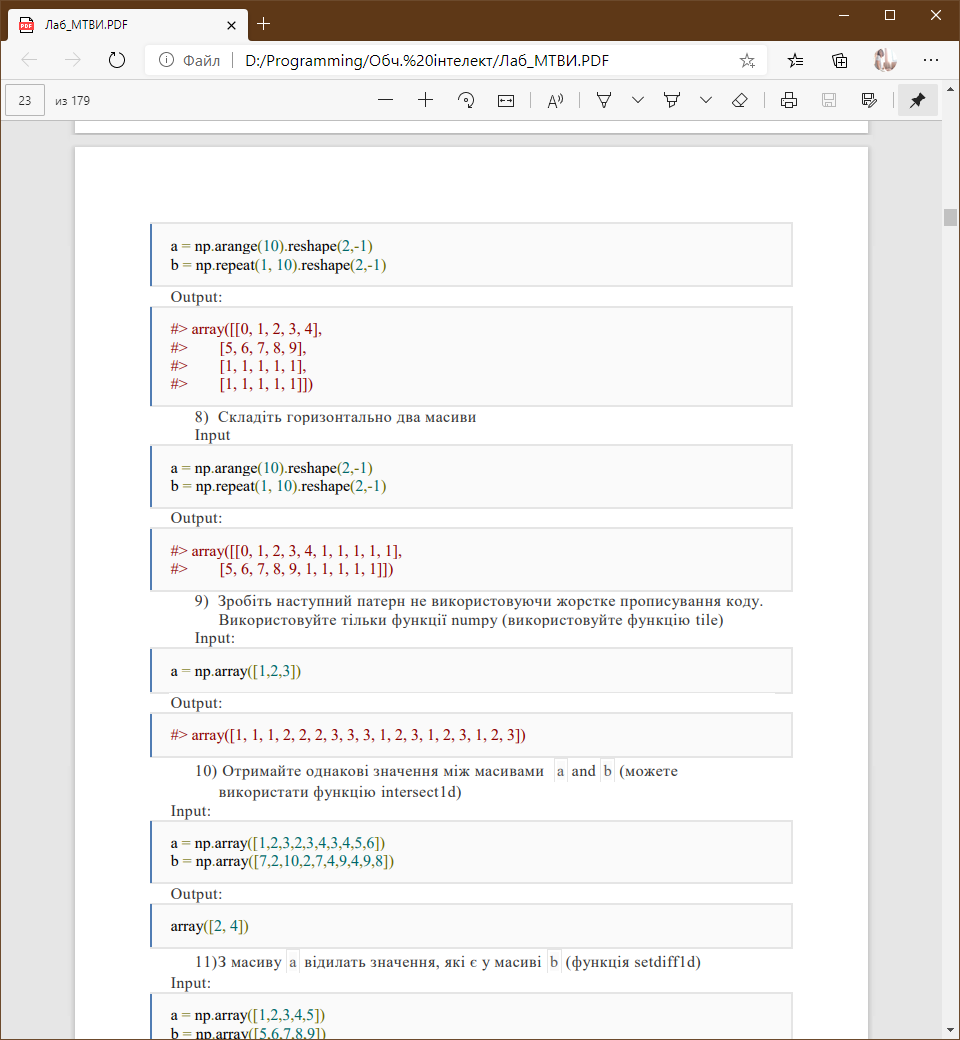


**Програмна реалізація:**

import numpy as np  
  
a = np.arange(10).reshape(2, -1)  
b = np.repeat(1, 10).reshape(2, -1)  
  
a = np.vstack((a, b))  
print(a)

**Результат:**

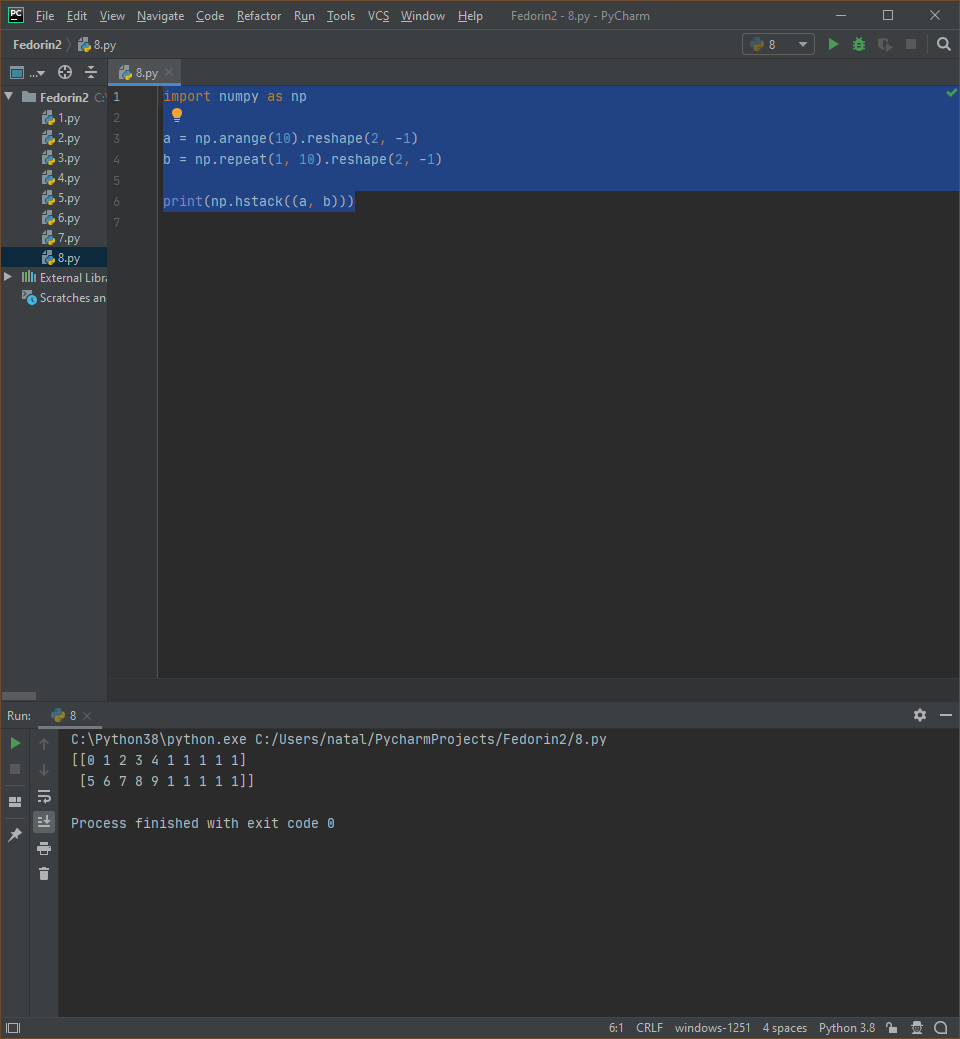


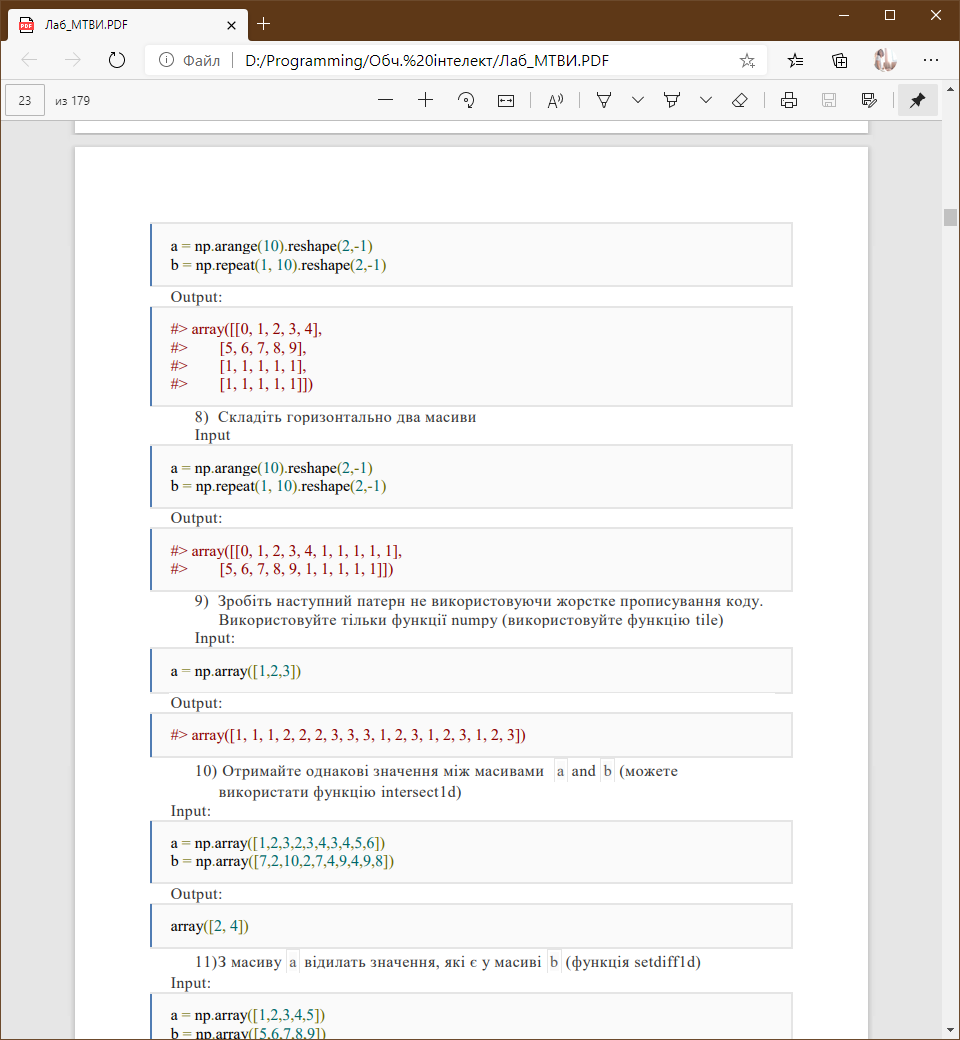


**Програмна реалізація:**

import numpy as np  
  
a = np.arange(10).reshape(2, -1)  
b = np.repeat(1, 10).reshape(2, -1)  
  
print(np.hstack((a, b)))

**Результат:**

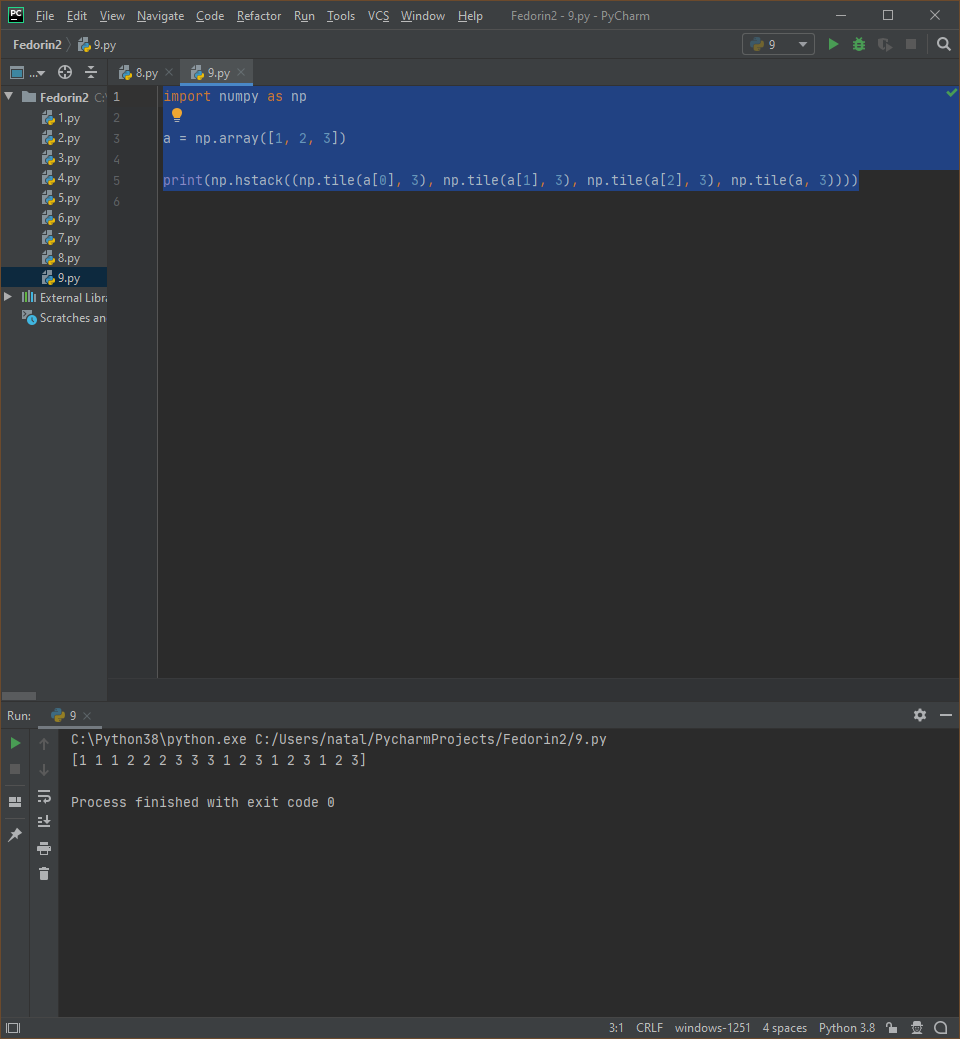


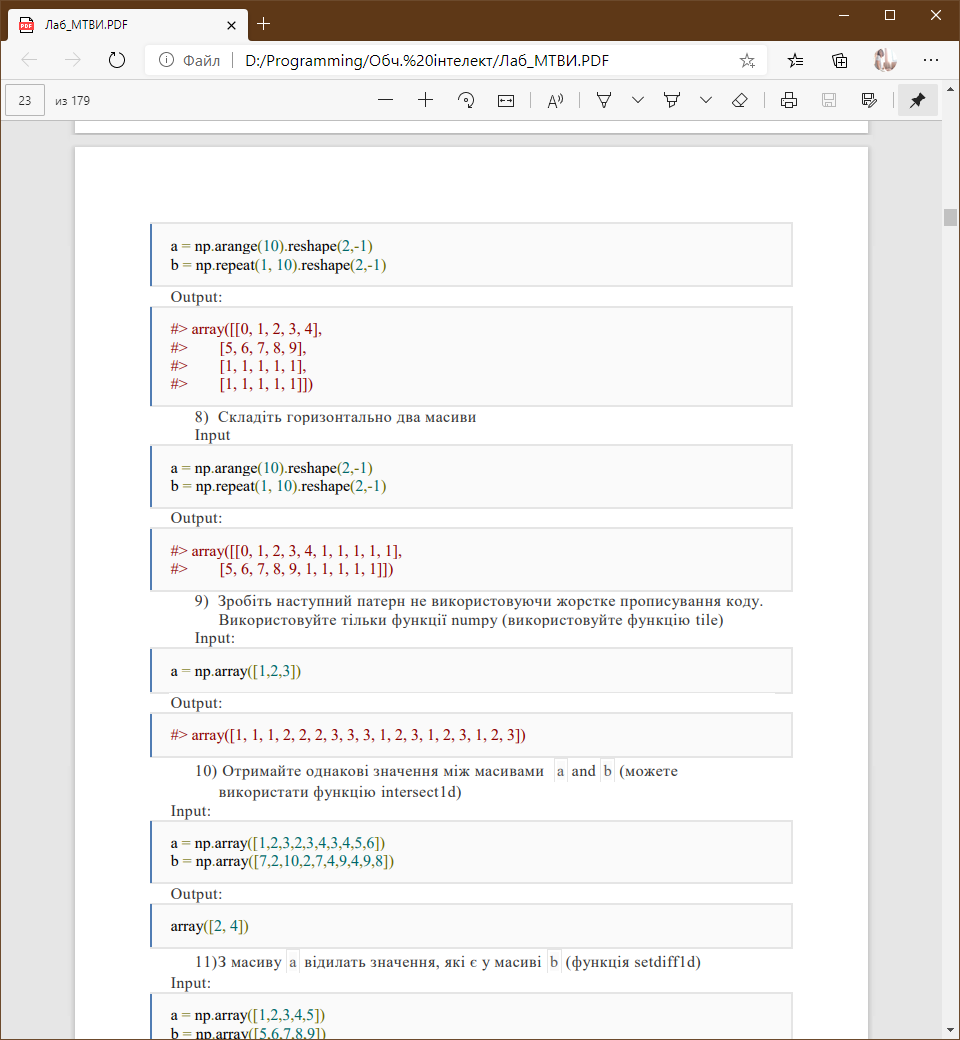


**Програмна реалізація:**

import numpy as np  
  
a = np.array([1, 2, 3])  
  
print(np.hstack((np.tile(a[0], 3), np.tile(a[1], 3), np.tile(a[2], 3), np.tile(a, 3))))

**Результат:**

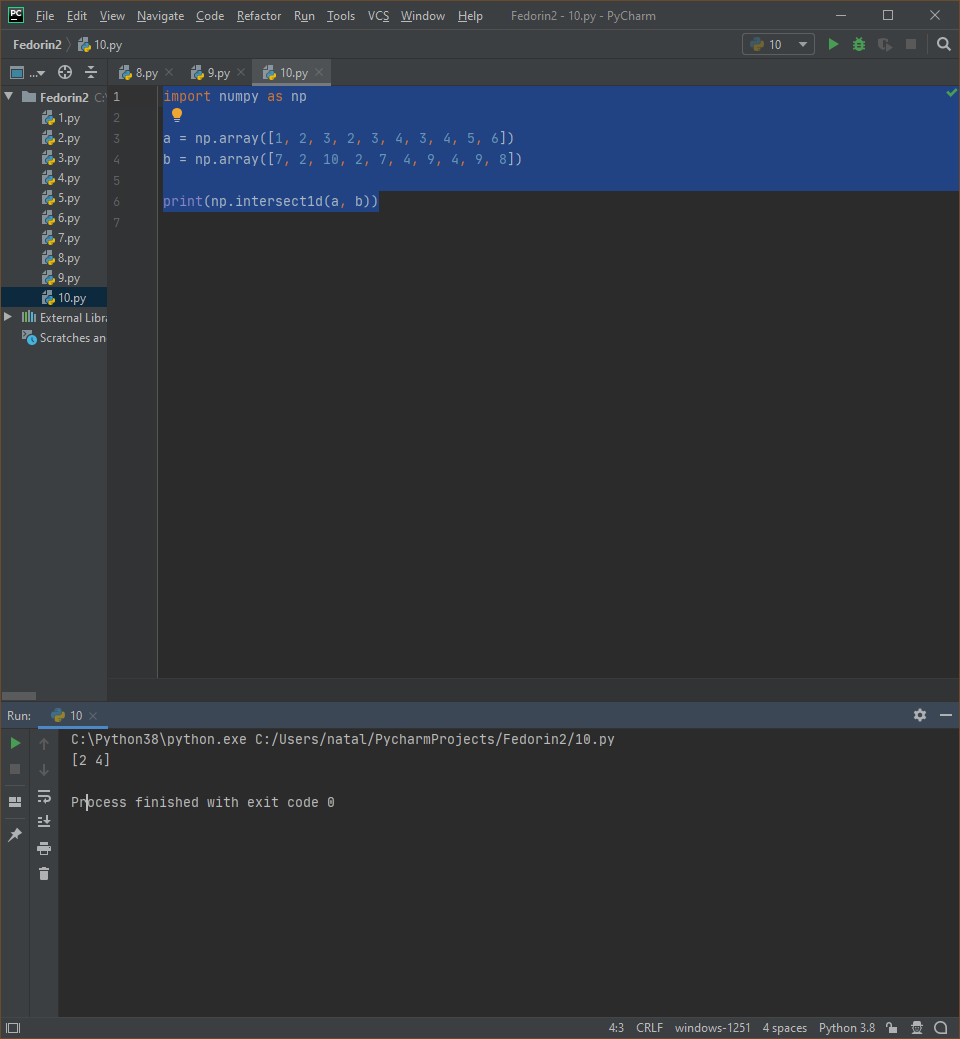


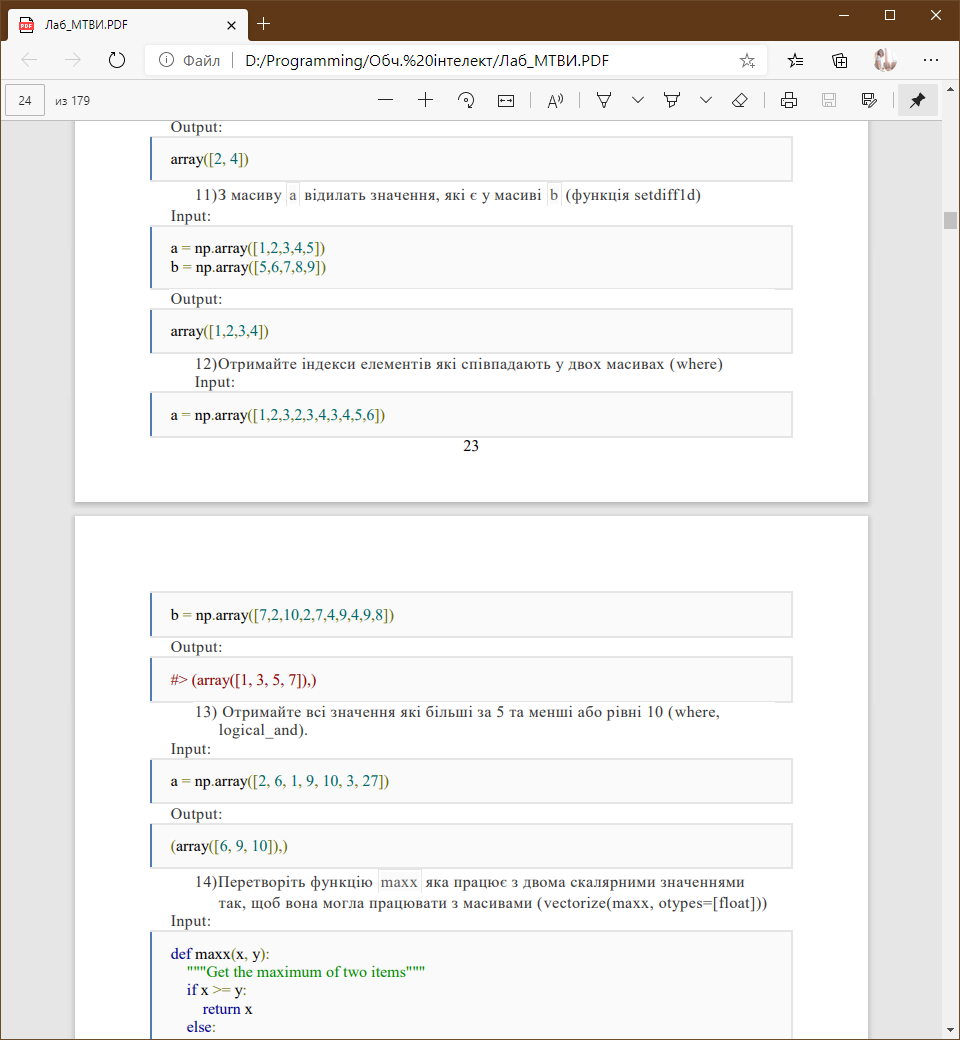


**Програмна реалізація:**

import numpy as np  
  
a = np.array([1, 2, 3, 2, 3, 4, 3, 4, 5, 6])  
b = np.array([7, 2, 10, 2, 7, 4, 9, 4, 9, 8])  
  
print(np.intersect1d(a, b))

**Результат:**

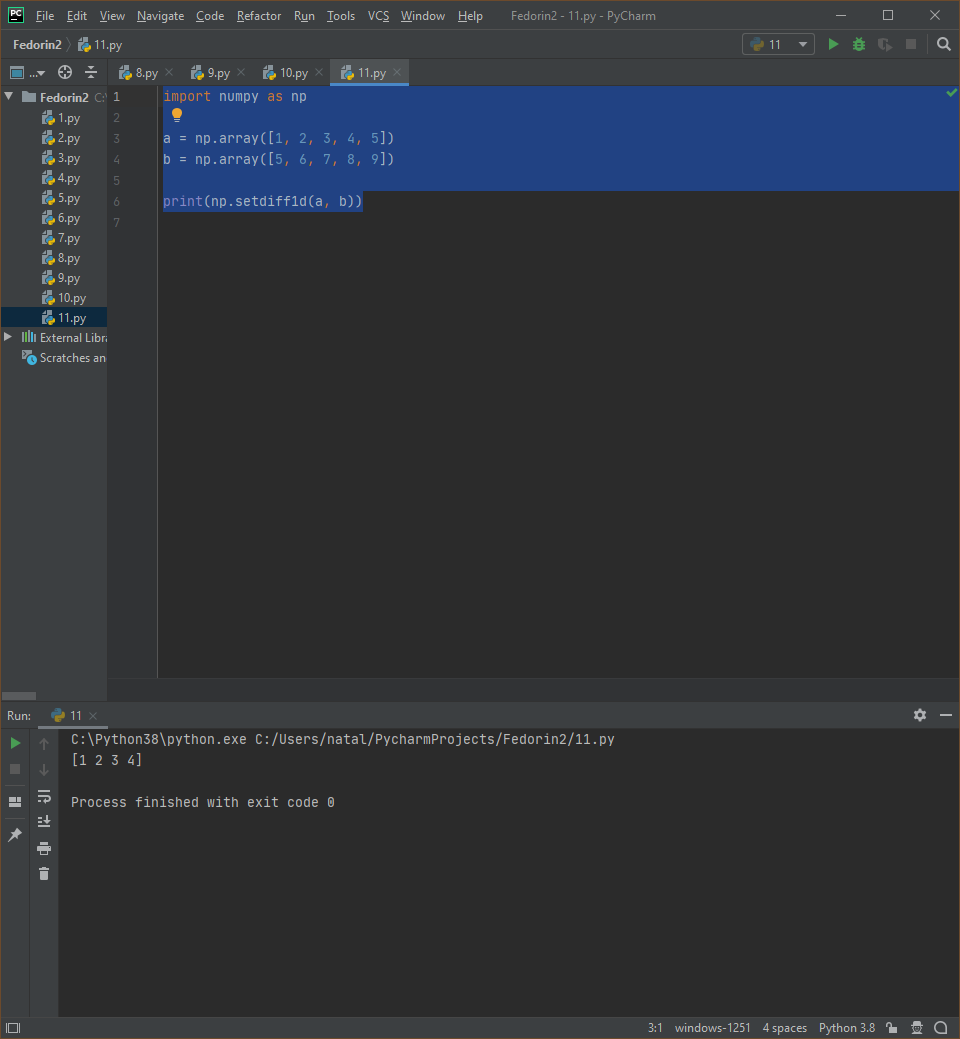


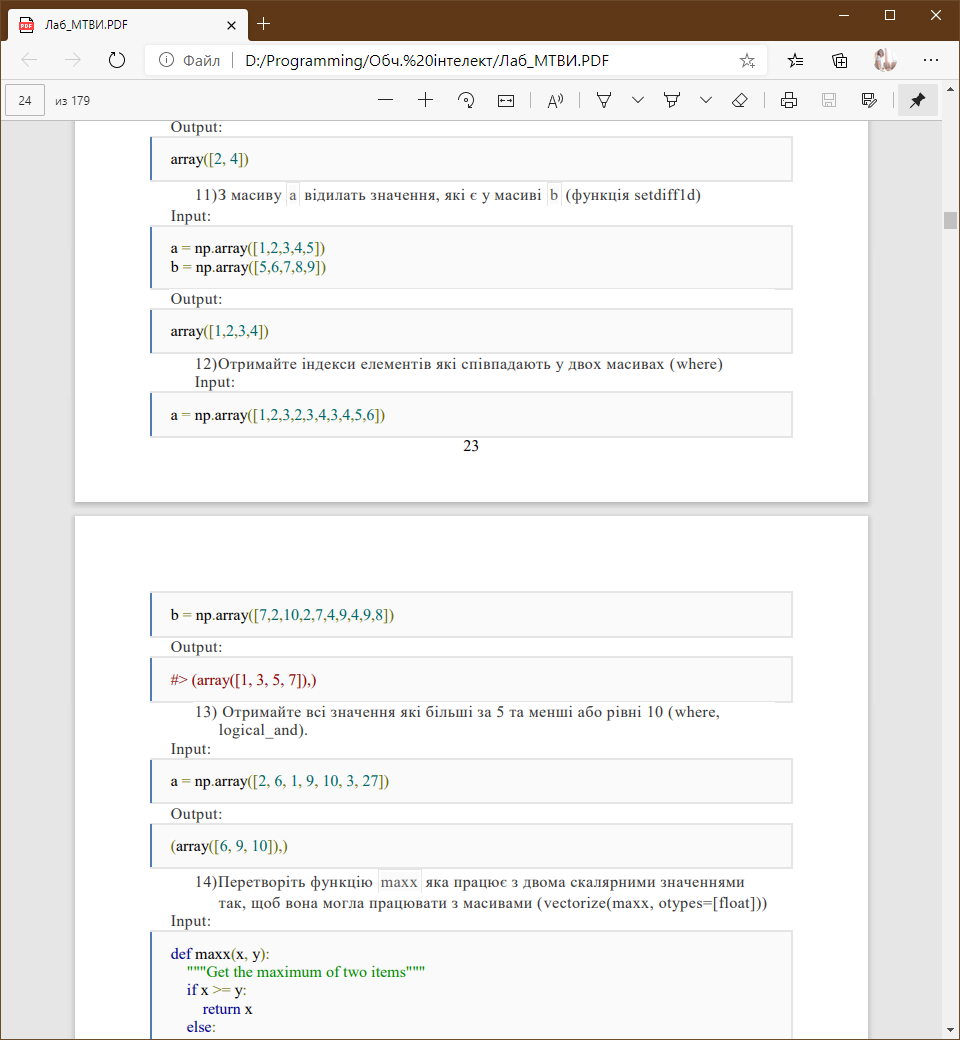


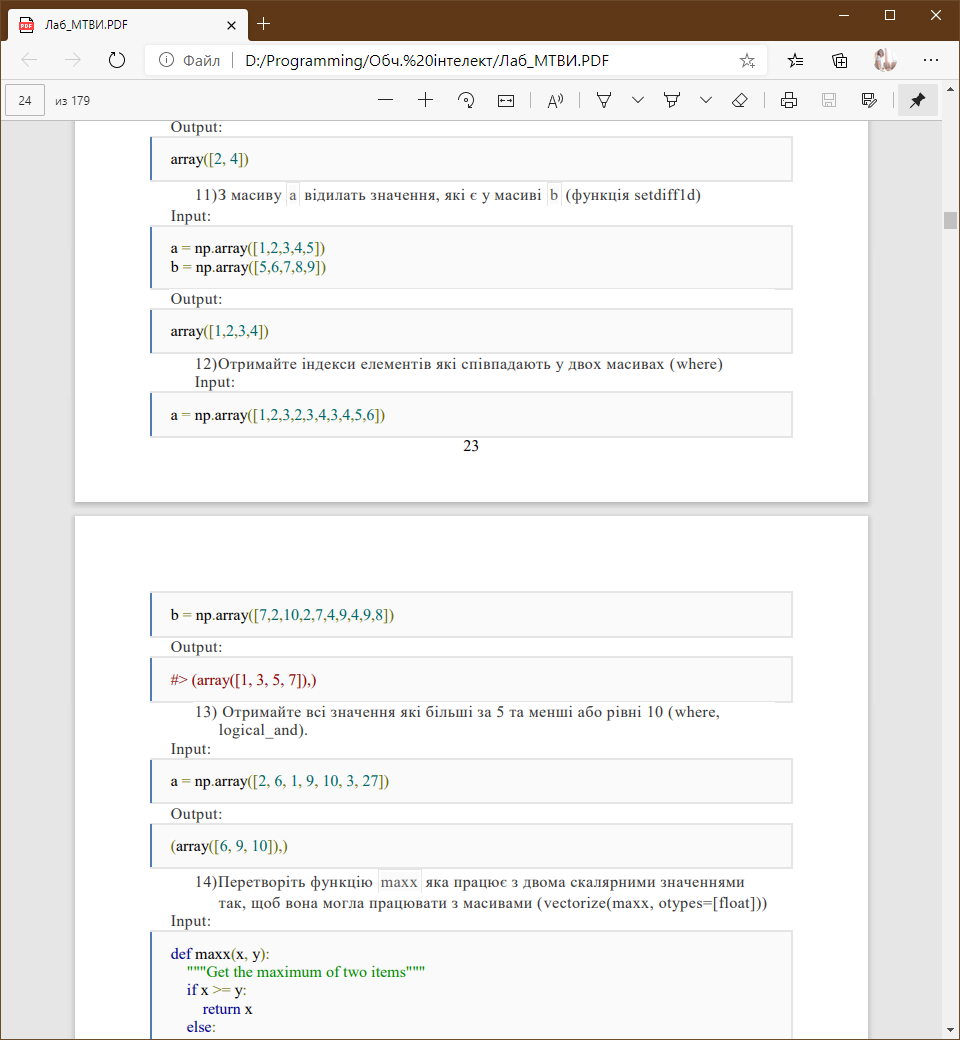
**Програмна реалізація:**

import numpy as np  
  
a = np.array([1, 2, 3, 4, 5])  
b = np.array([5, 6, 7, 8, 9])  
  
print(np.setdiff1d(a, b))

**Результат:**



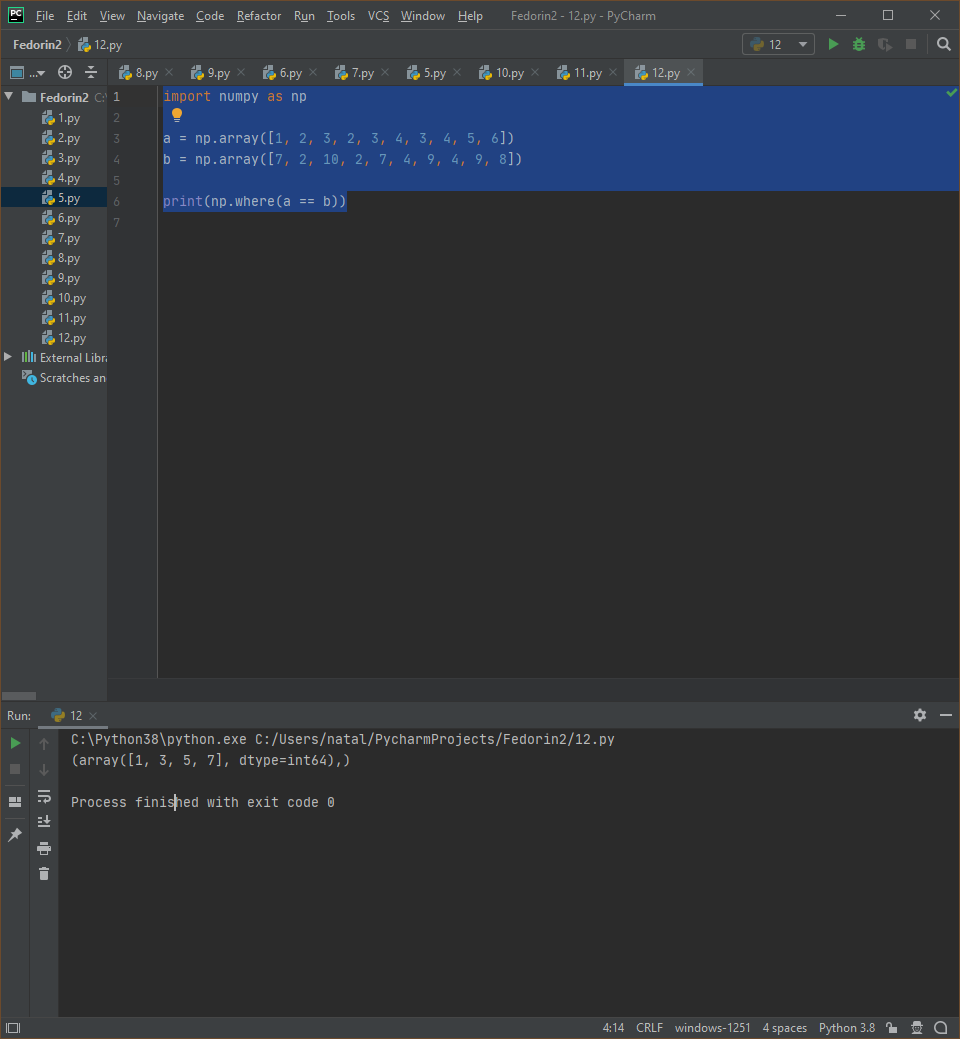


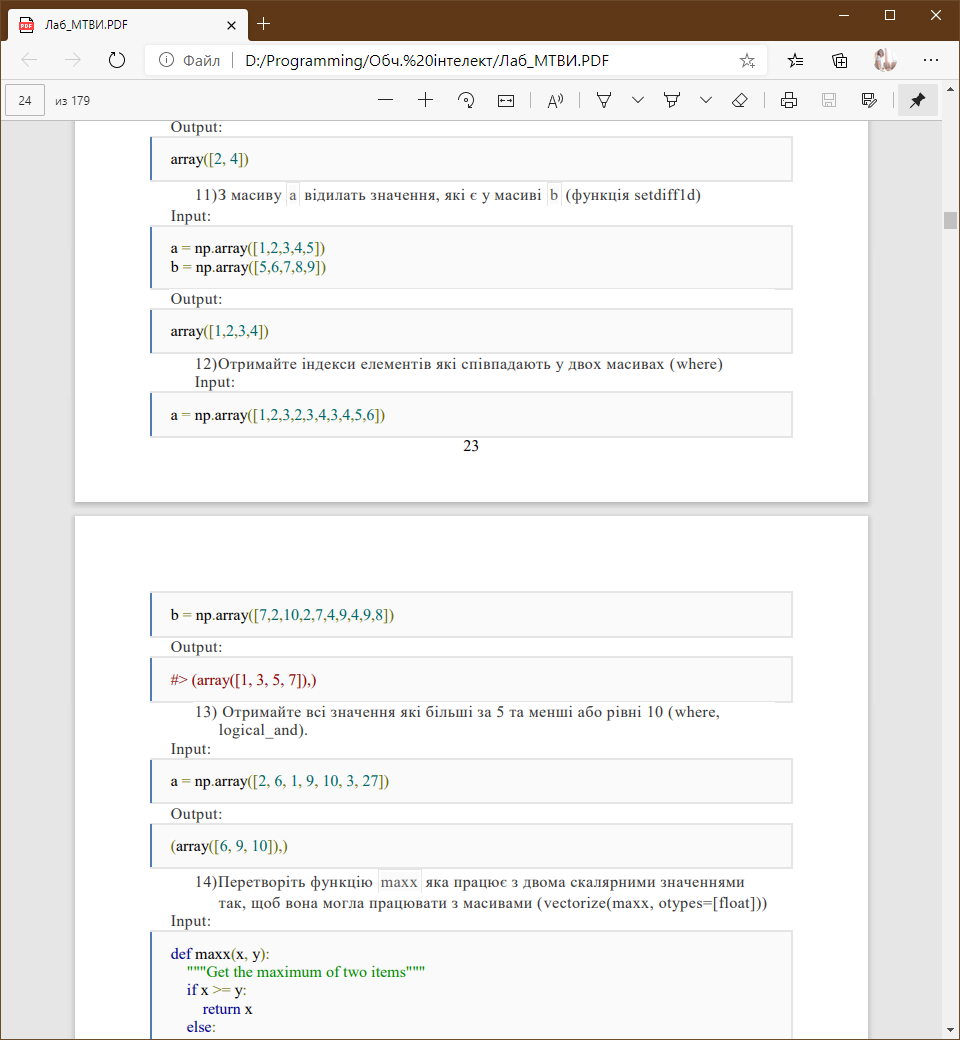


**Програмна реалізація:**

import numpy as np  
  
a = np.array([1, 2, 3, 2, 3, 4, 3, 4, 5, 6])  
b = np.array([7, 2, 10, 2, 7, 4, 9, 4, 9, 8])  
  
print(np.where(a == b))

**Результат:**

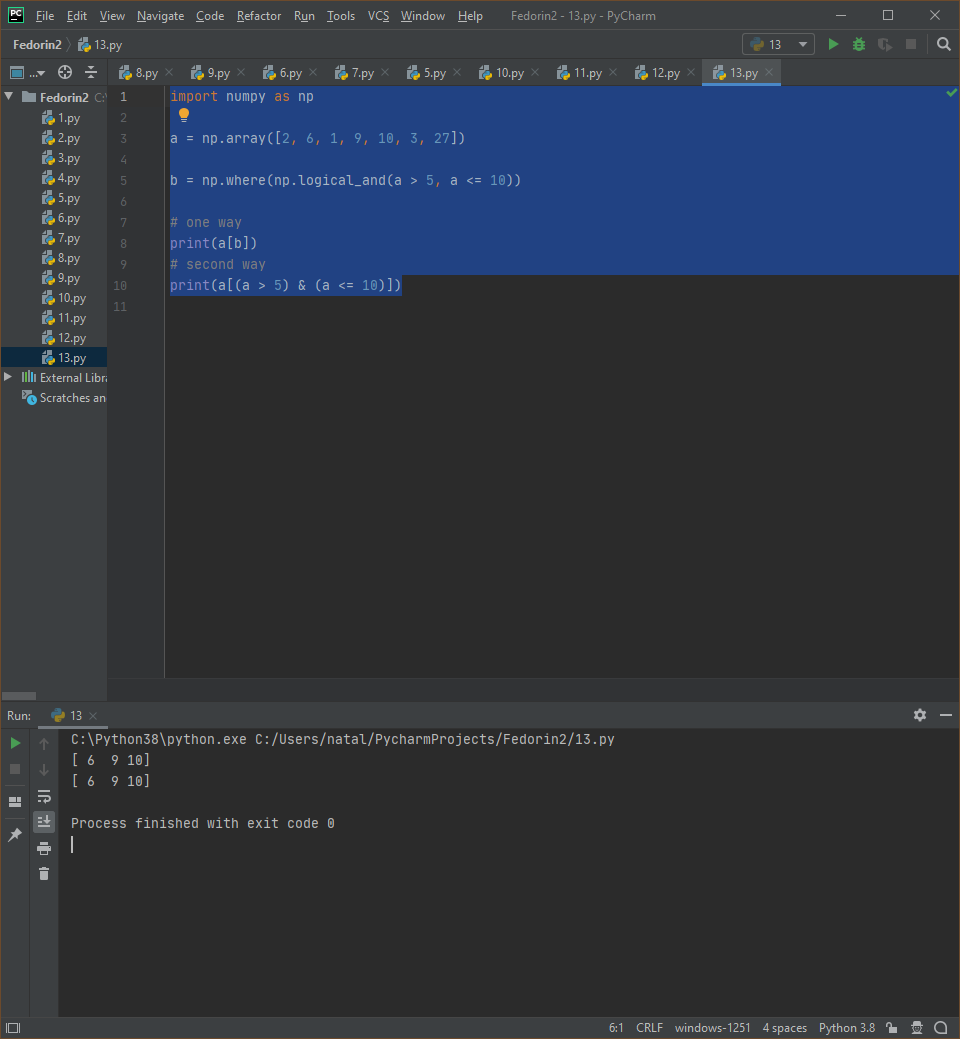


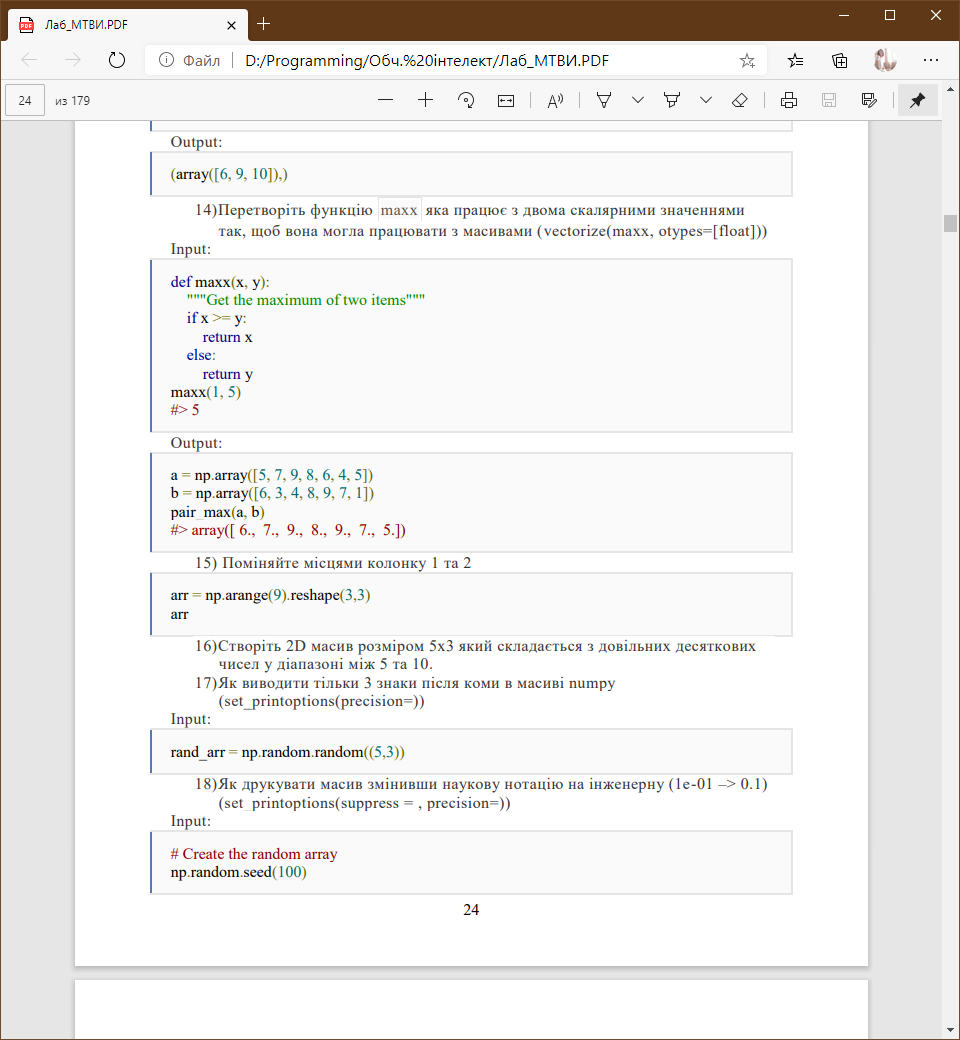


**Програмна реалізація:**

import numpy as np  
  
a = np.array([2, 6, 1, 9, 10, 3, 27])  
  
b = np.where(np.logical\_and(a > 5, a <= 10))  
  
# one way  
print(a[b])  
# second way  
print(a[(a > 5) & (a <= 10)])

**Результат:**

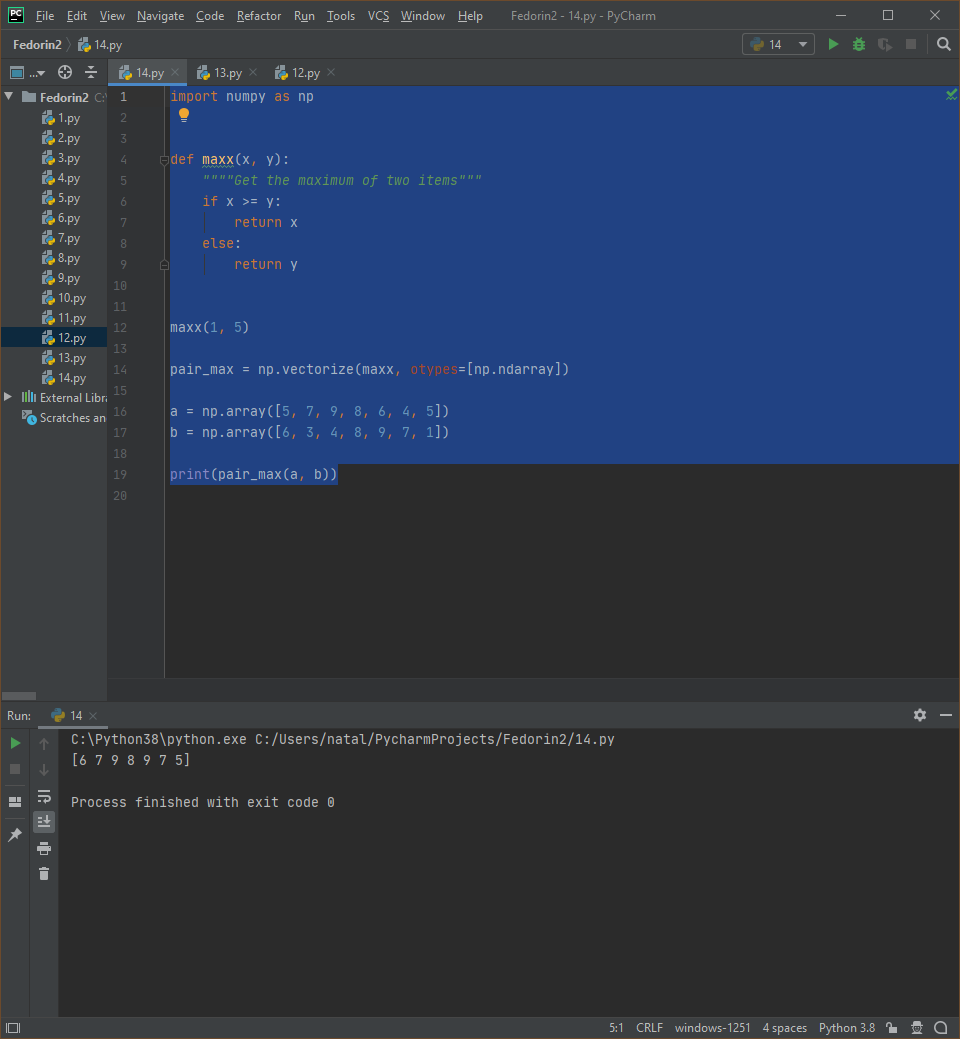


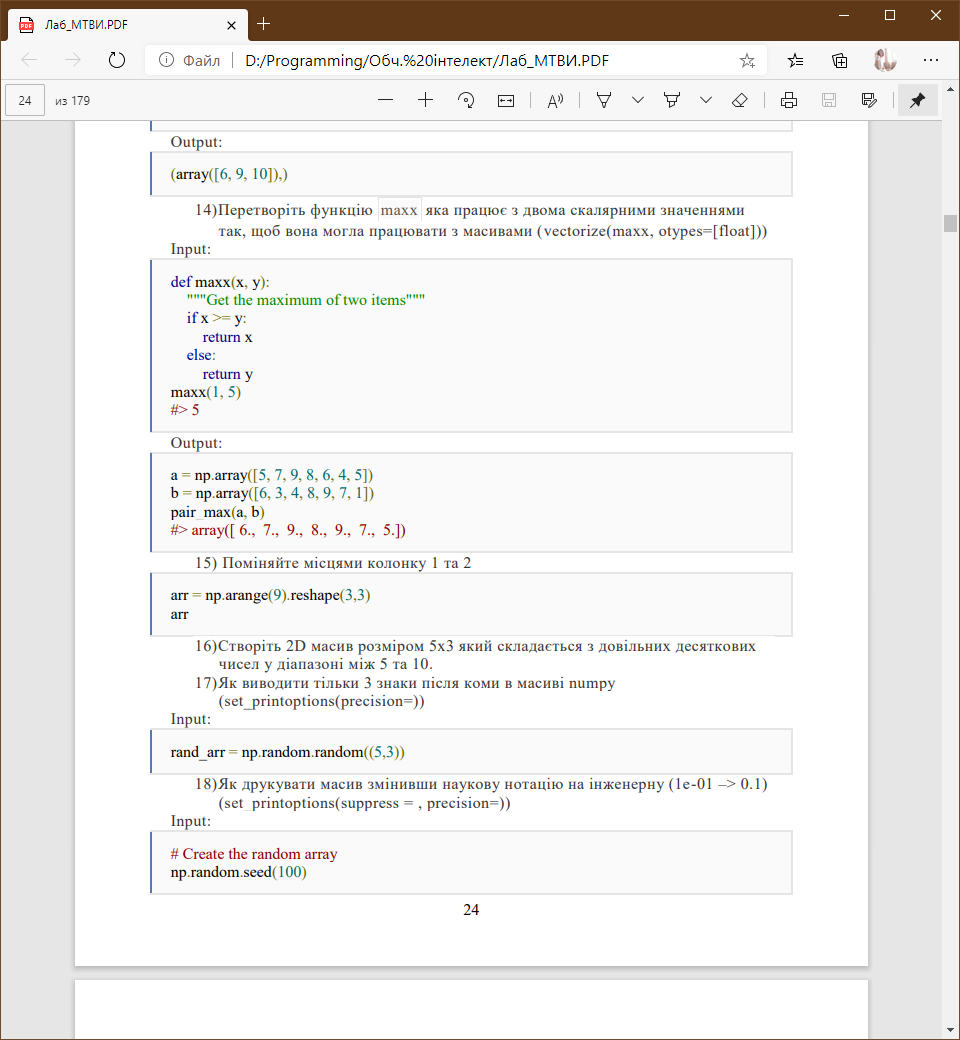


**Програмна реалізація:**

import numpy as np  
  
  
def maxx(x, y):  
 *""""Get the maximum of two items"""* if x >= y:  
 return x  
 else:  
 return y  
  
  
maxx(1, 5)  
  
pair\_max = np.vectorize(maxx, otypes=[np.ndarray])  
  
a = np.array([5, 7, 9, 8, 6, 4, 5])  
b = np.array([6, 3, 4, 8, 9, 7, 1])  
  
print(pair\_max(a, b))

**Результат:**

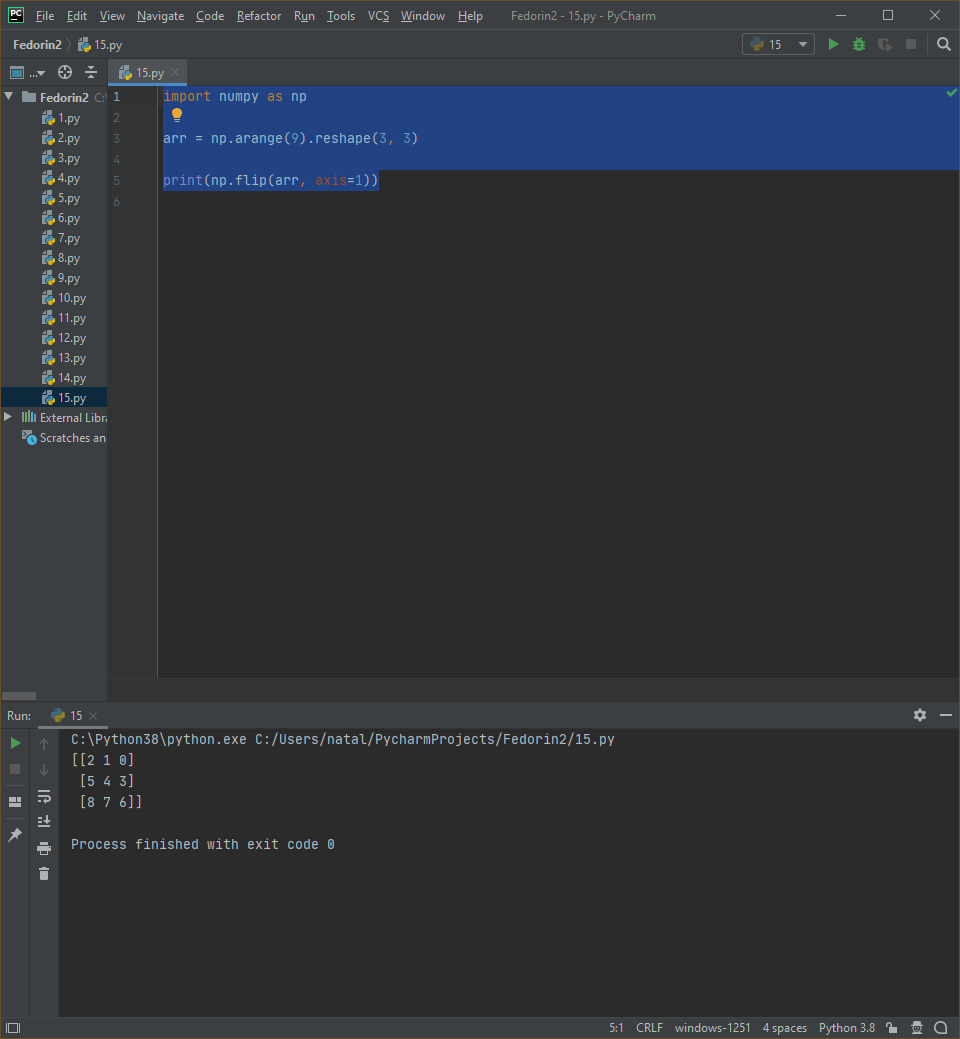


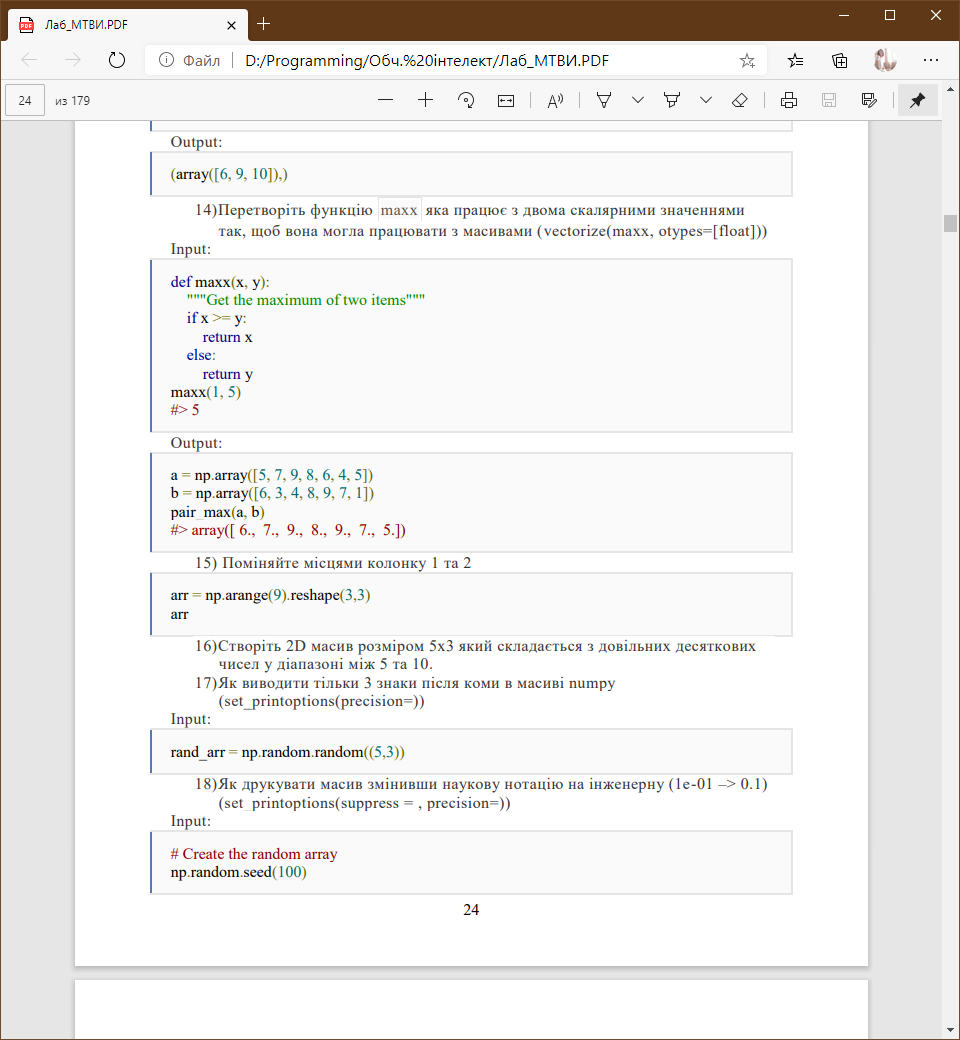


**Програмна реалізація:**

import numpy as np  
arr = np.arange(9).reshape(3, 3)  
print(np.flip(arr, axis=1))

**Результат:**

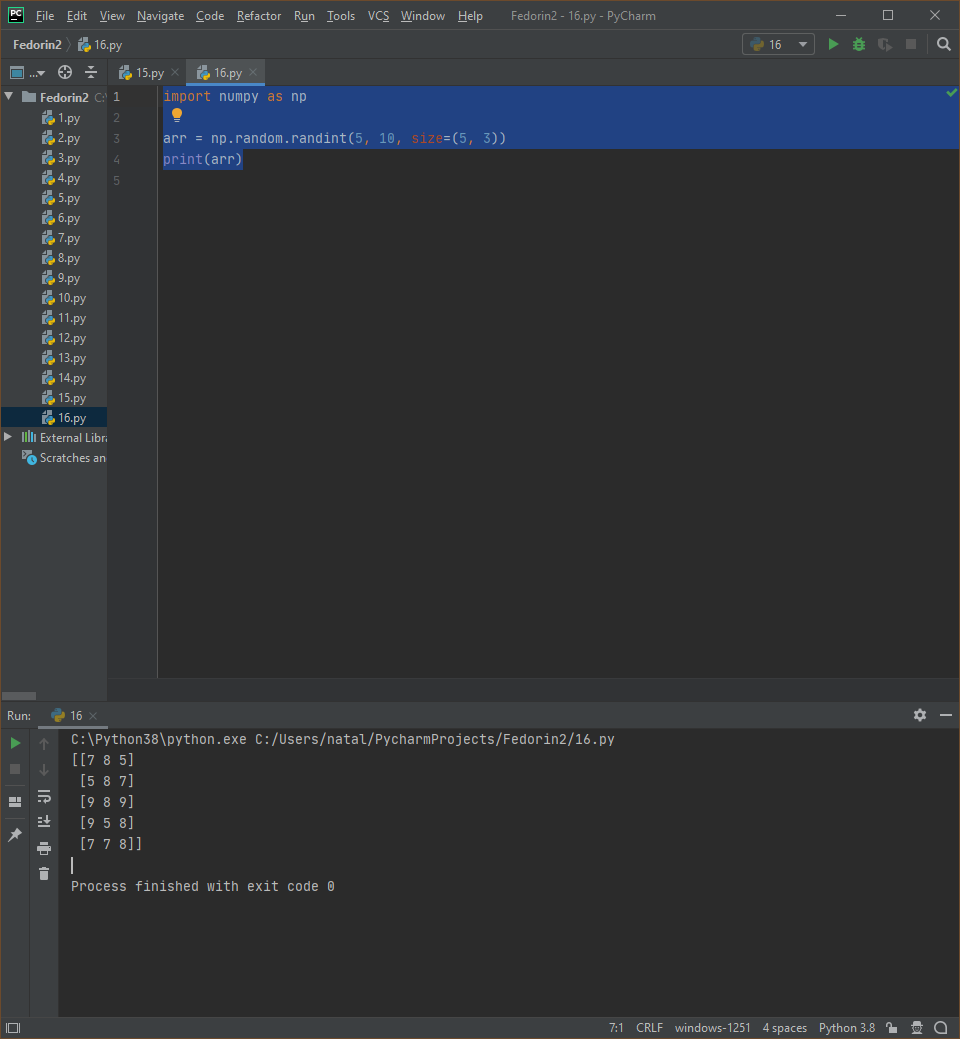


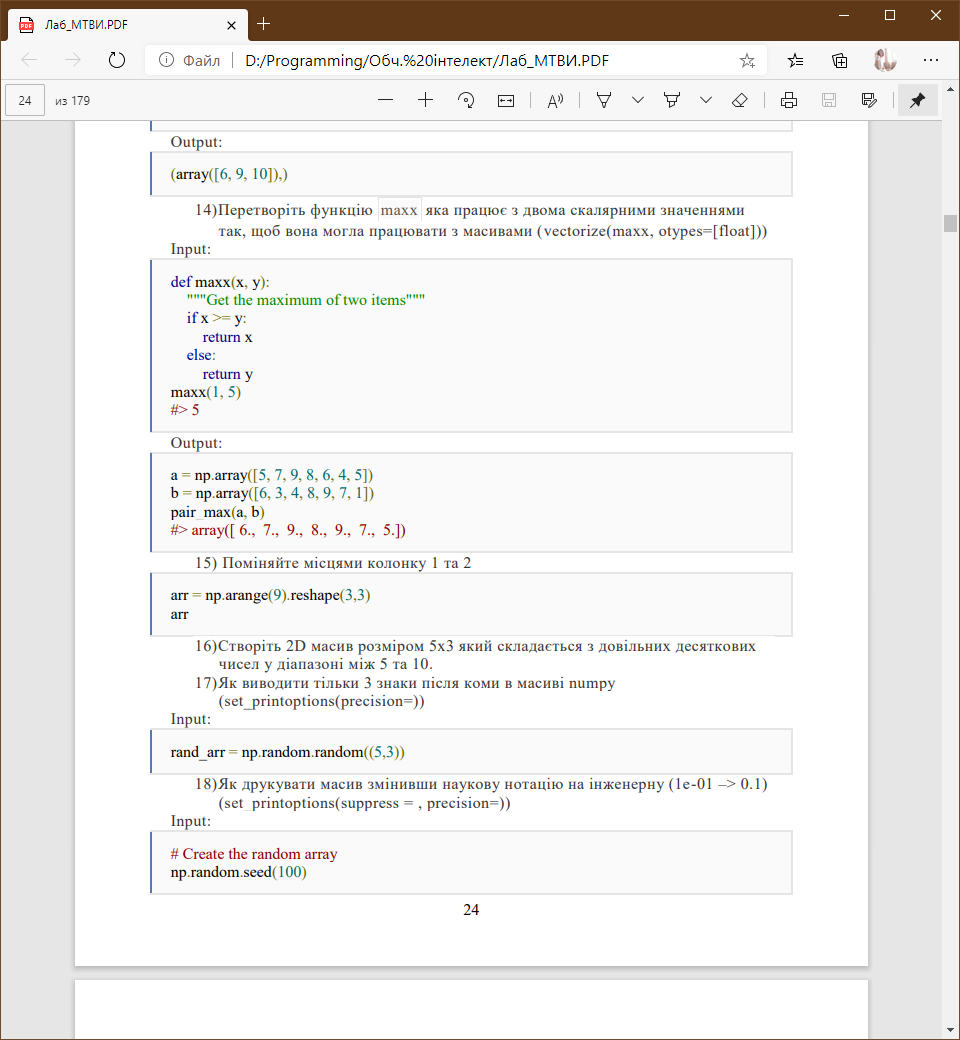


**Програмна реалізація:**

import numpy as np  
  
arr = np.random.randint(5, 10, size=(5, 3))  
print(arr)

**Результат:**

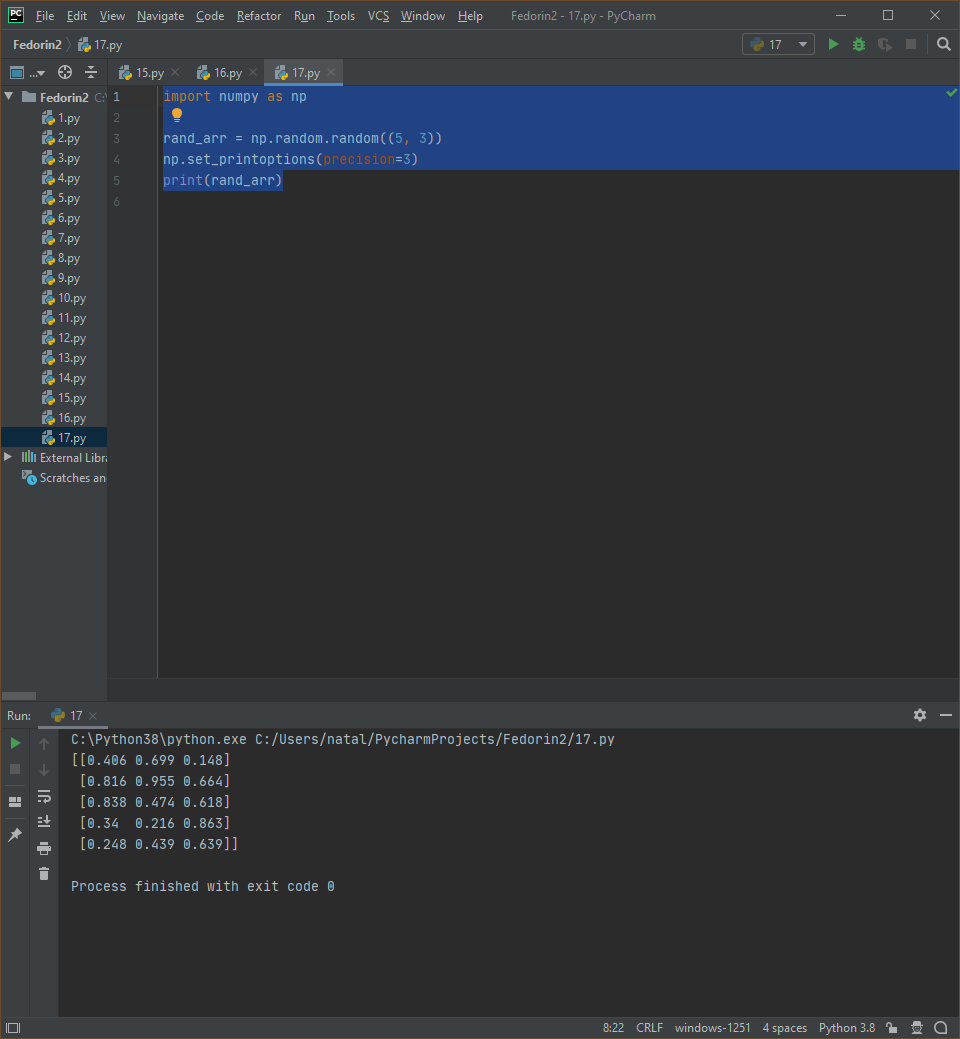


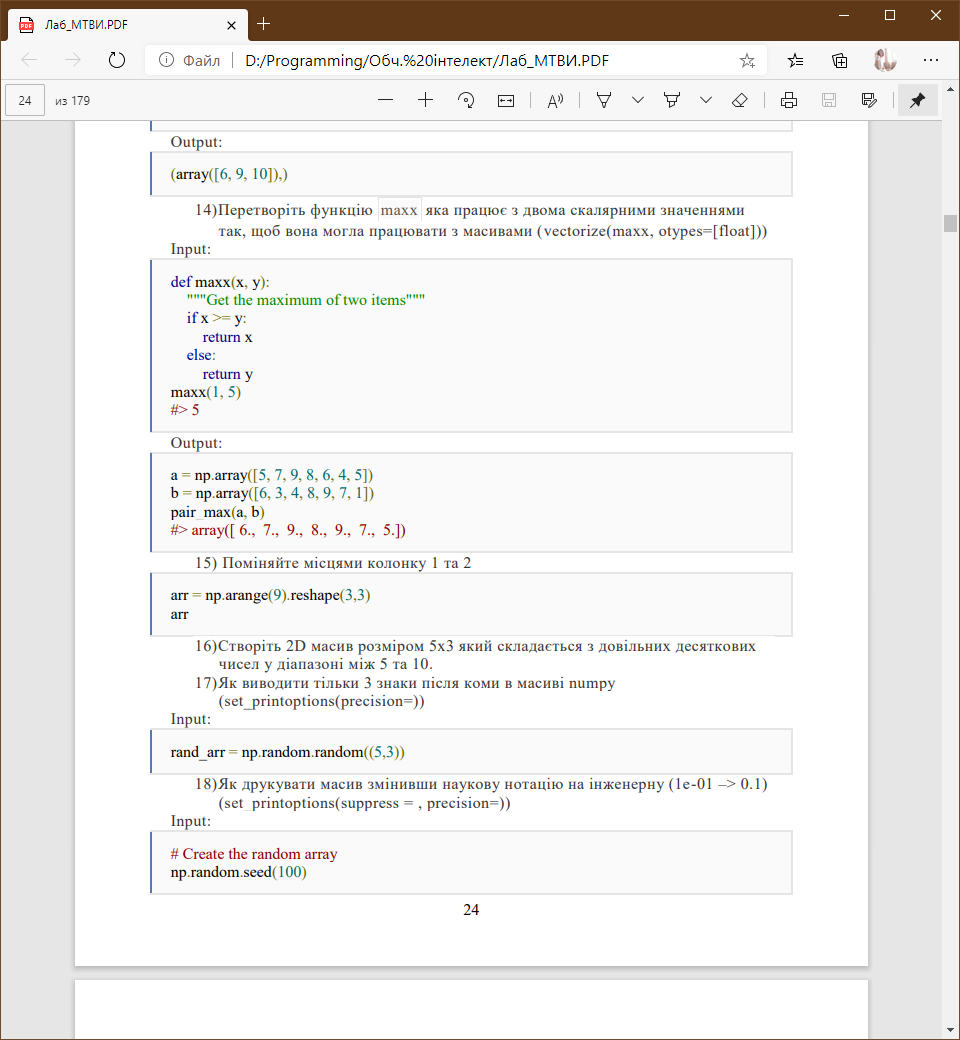


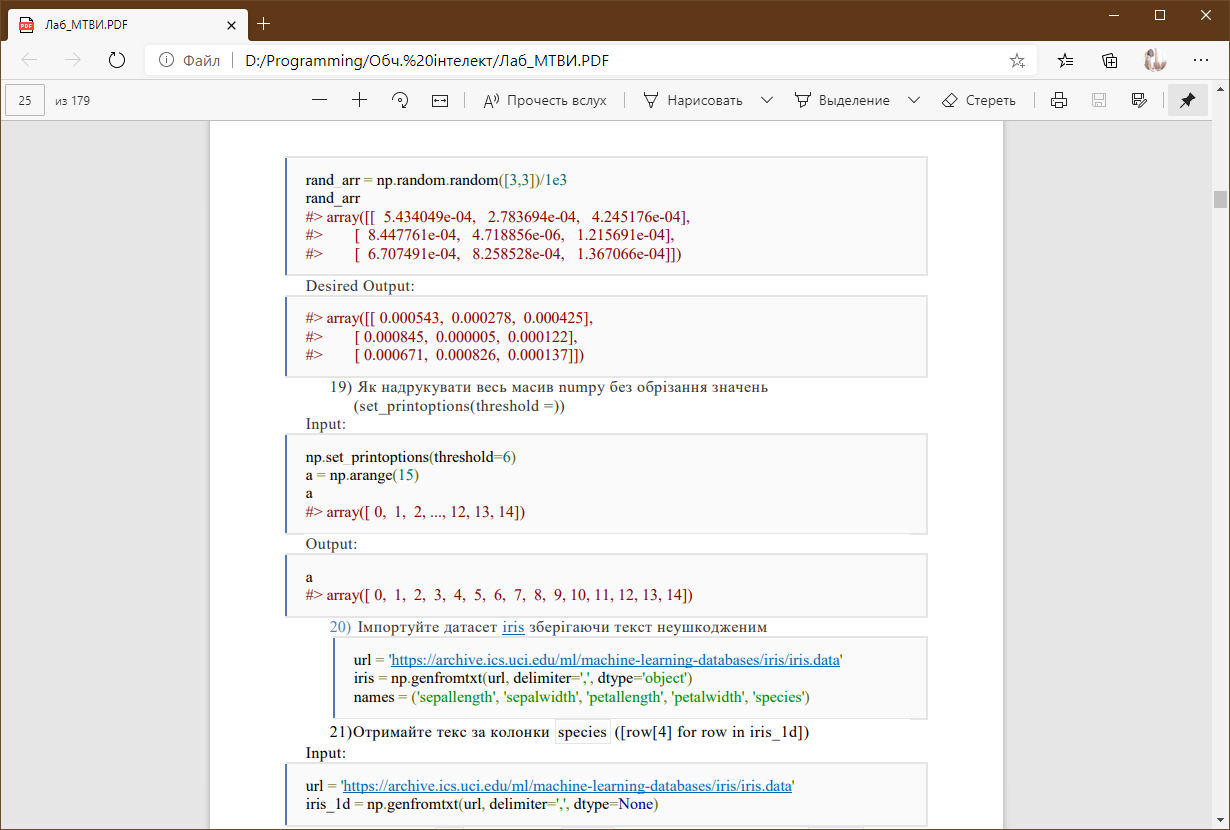
**Програмна реалізація:**

import numpy as np  
  
rand\_arr = np.random.random((5, 3))  
np.set\_printoptions(precision=3)  
print(rand\_arr)

**Результат:**



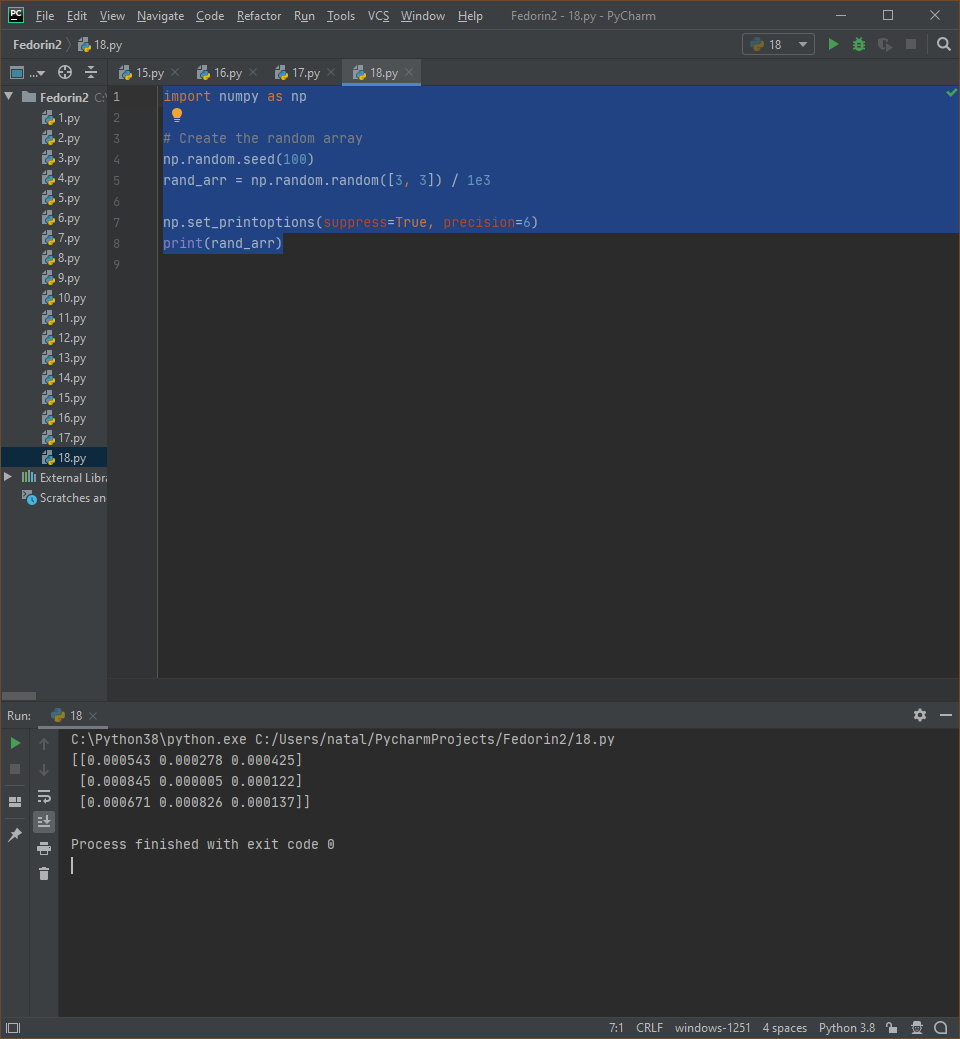


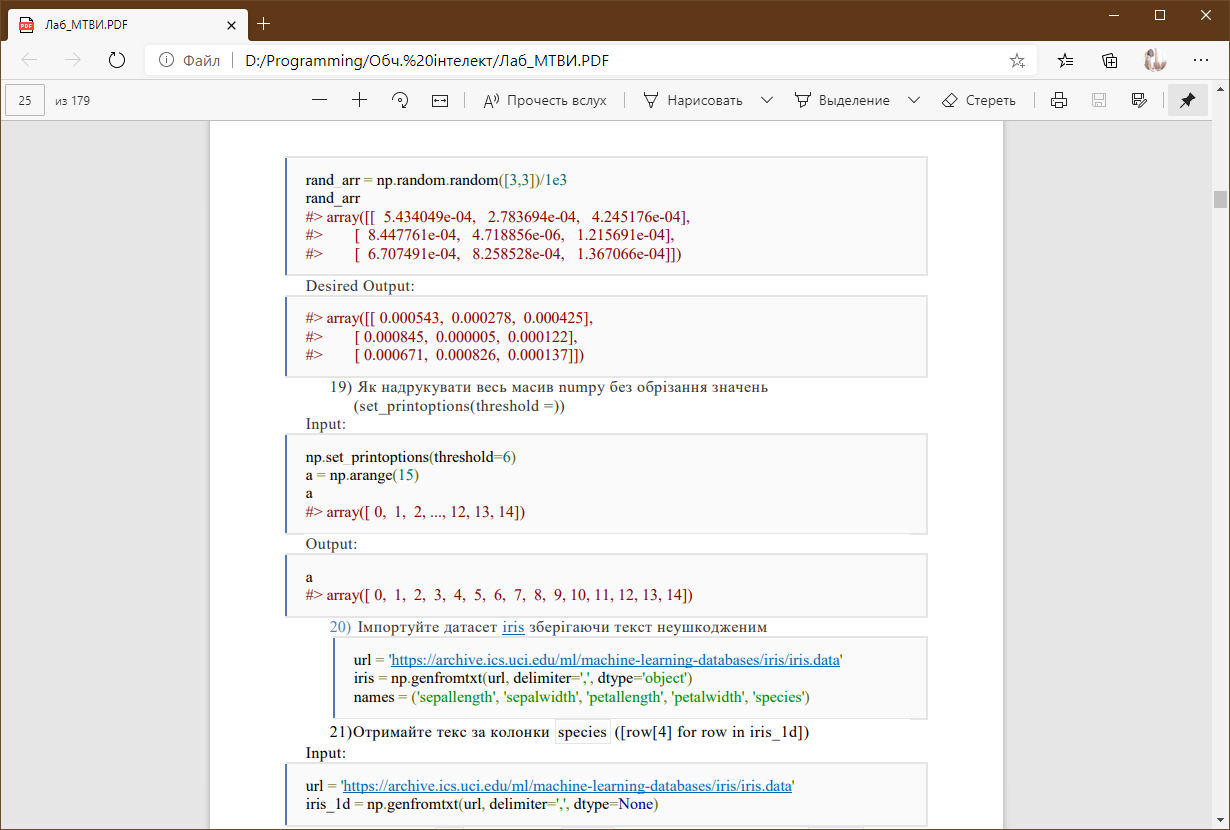


**Програмна реалізація:**

import numpy as np  
  
# Create the random array  
np.random.seed(100)  
rand\_arr = np.random.random([3, 3]) / 1e3  
  
np.set\_printoptions(suppress=True, precision=6)  
print(rand\_arr)

**Результат:**

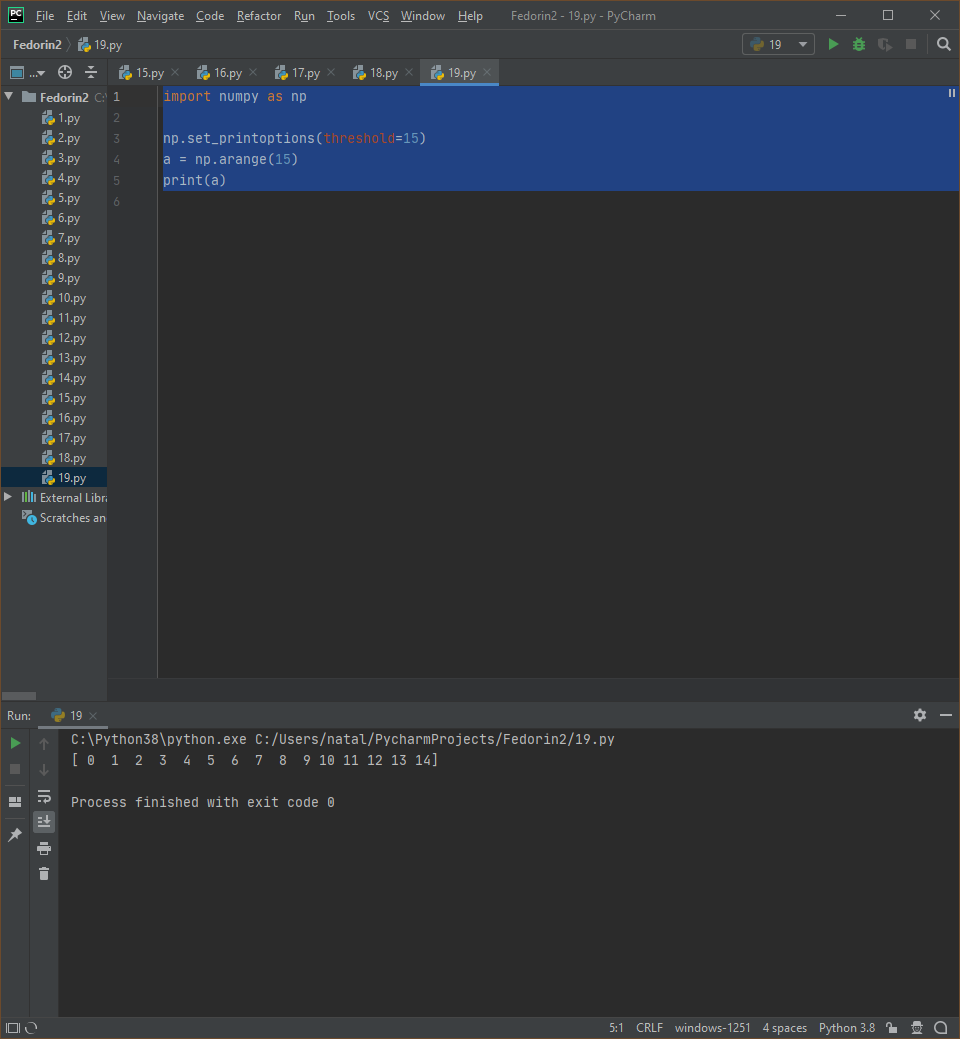


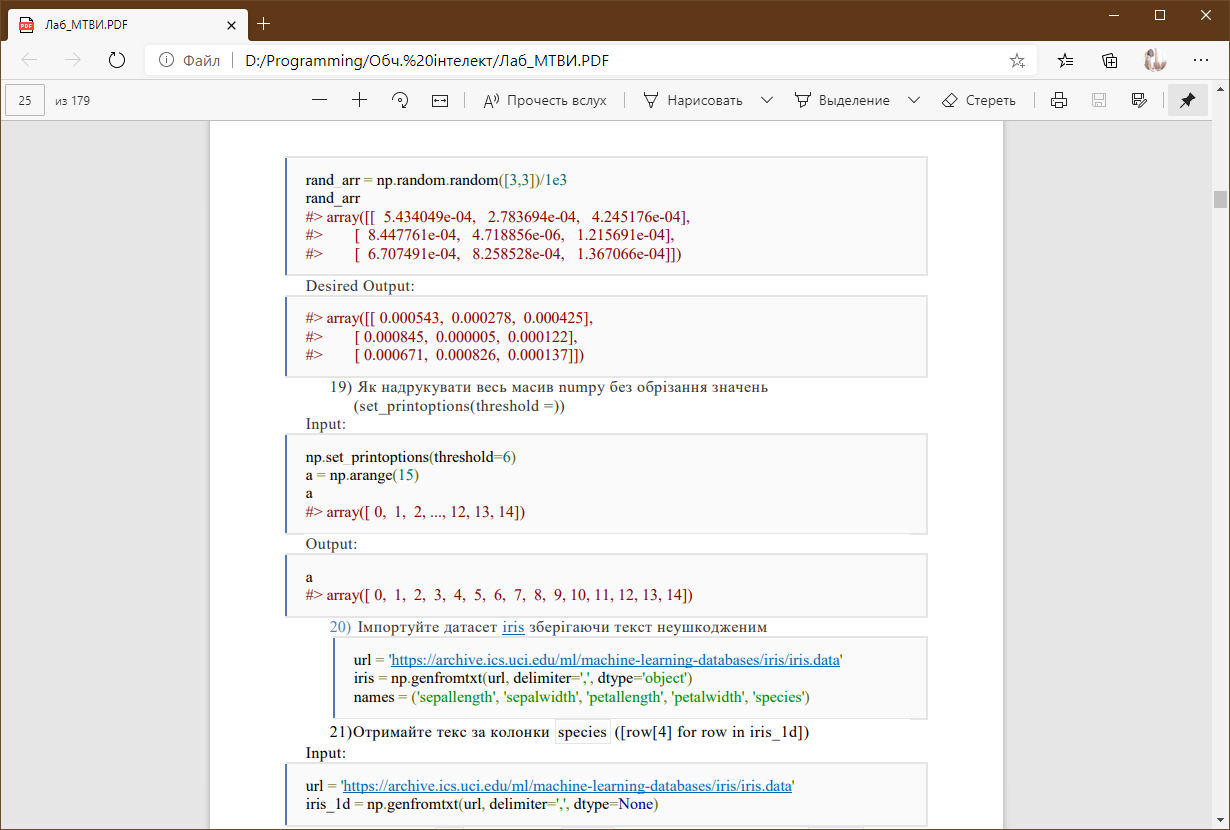


**Програмна реалізація:**

import numpy as np  
  
np.set\_printoptions(threshold=15)  
a = np.arange(15)  
print(a)

**Результат:**

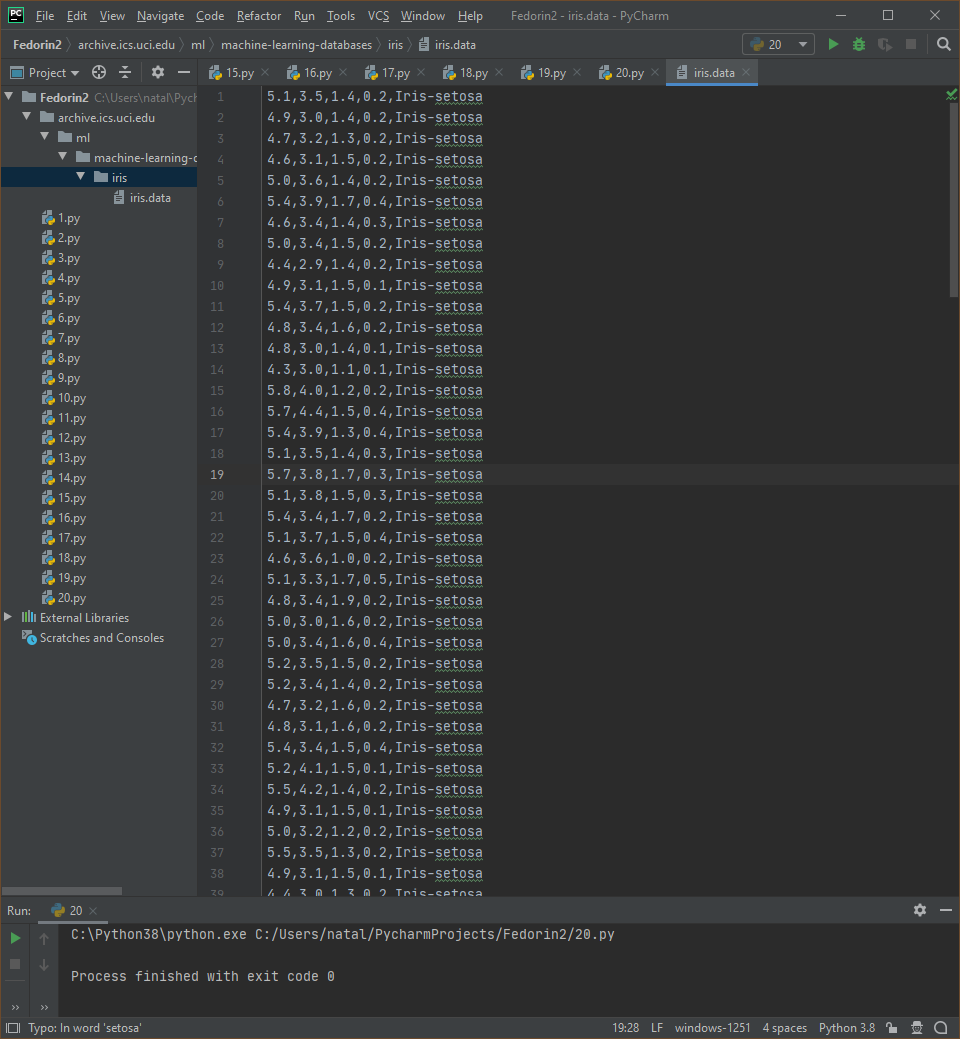


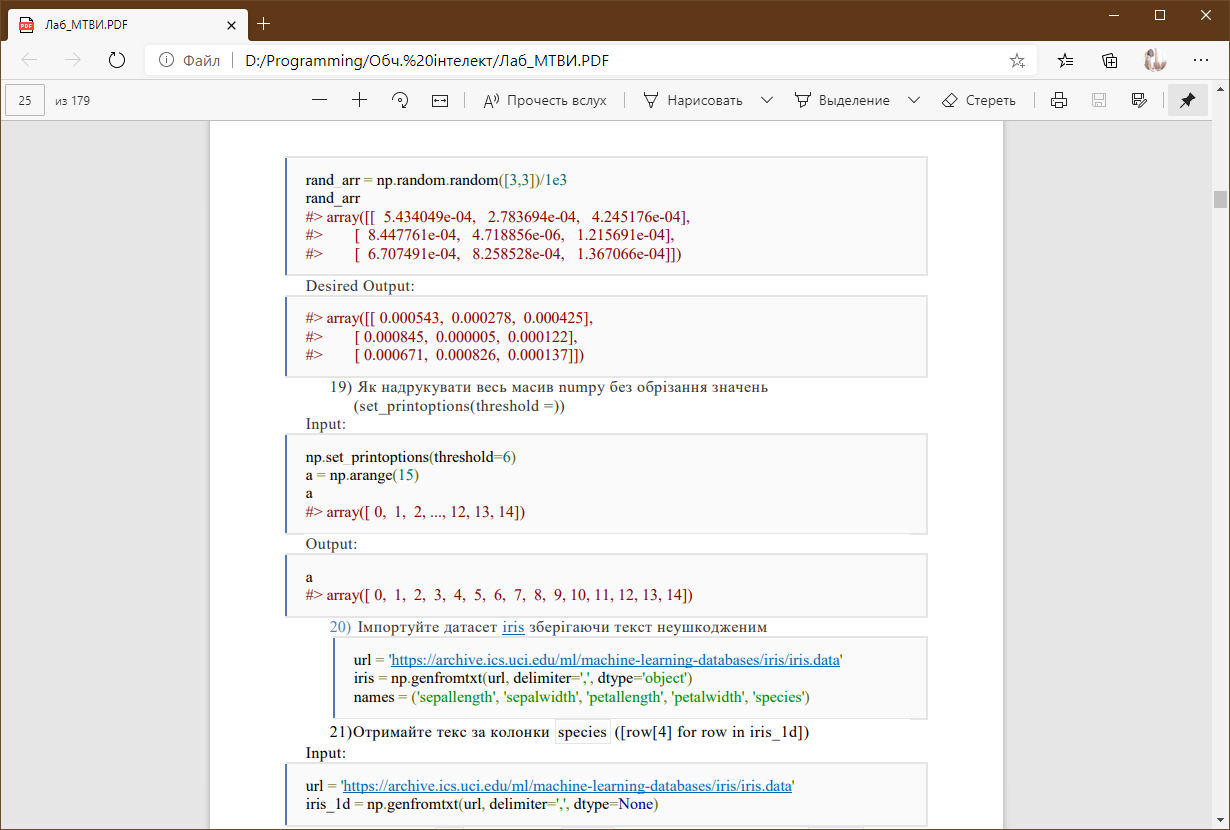


**Програмна реалізація:**

import numpy as np  
  
np.set\_printoptions(threshold=15)  
a = np.arange(15)  
print(a)

**Результат:**

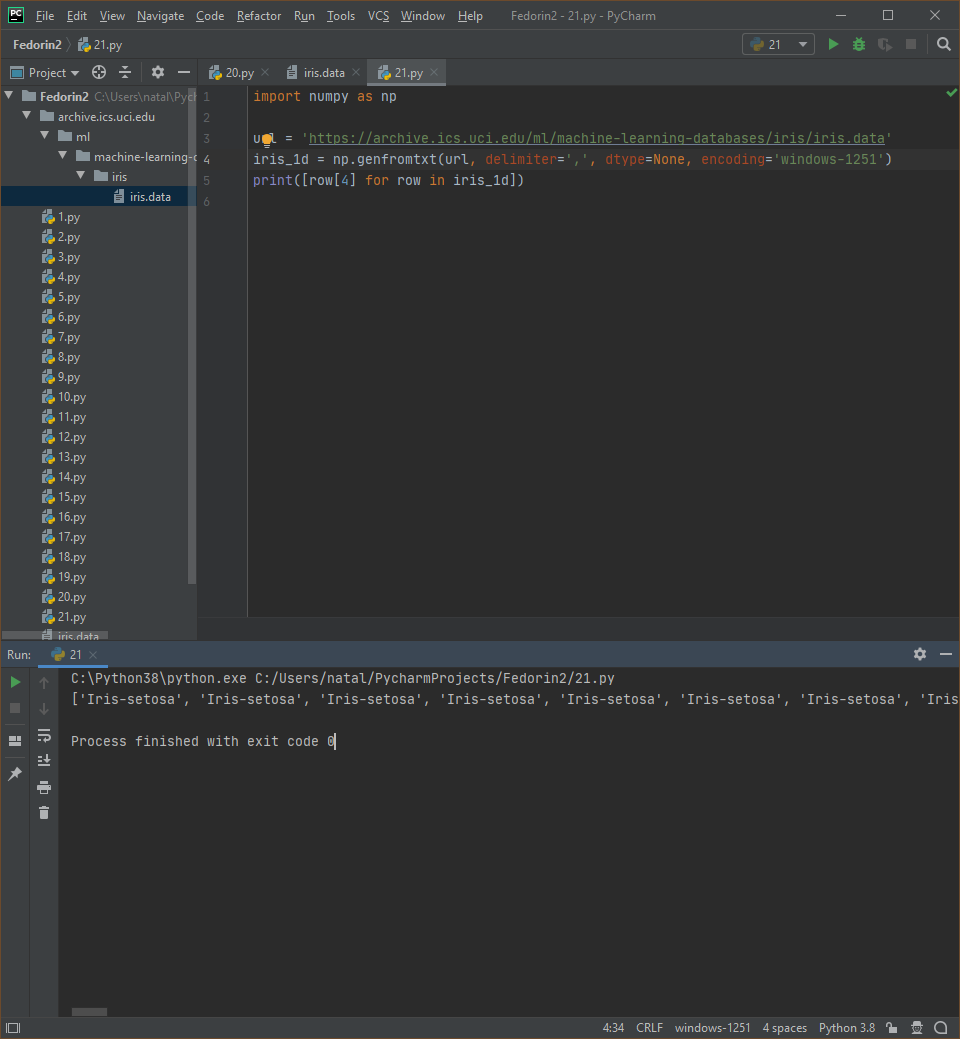


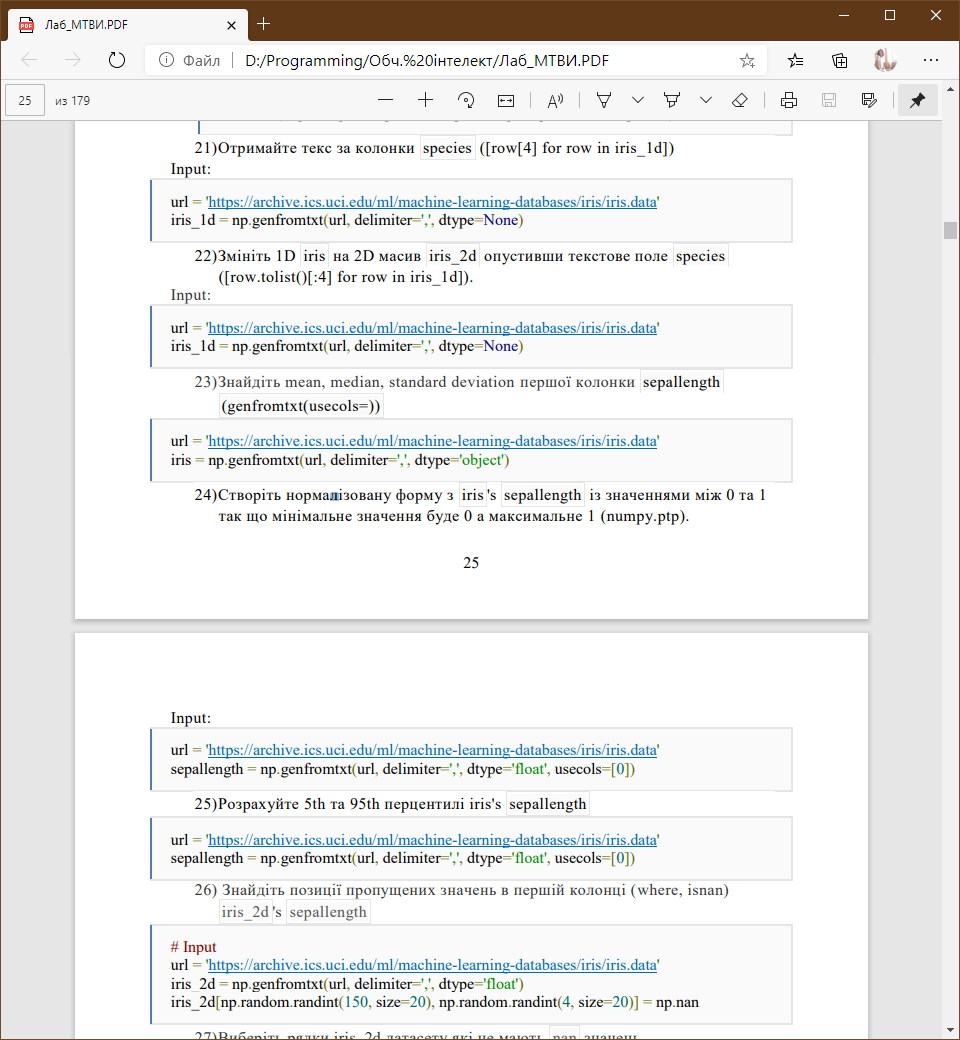


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None, encoding='windows-1251')  
print([row[4] for row in iris\_1d])

**Результат:**

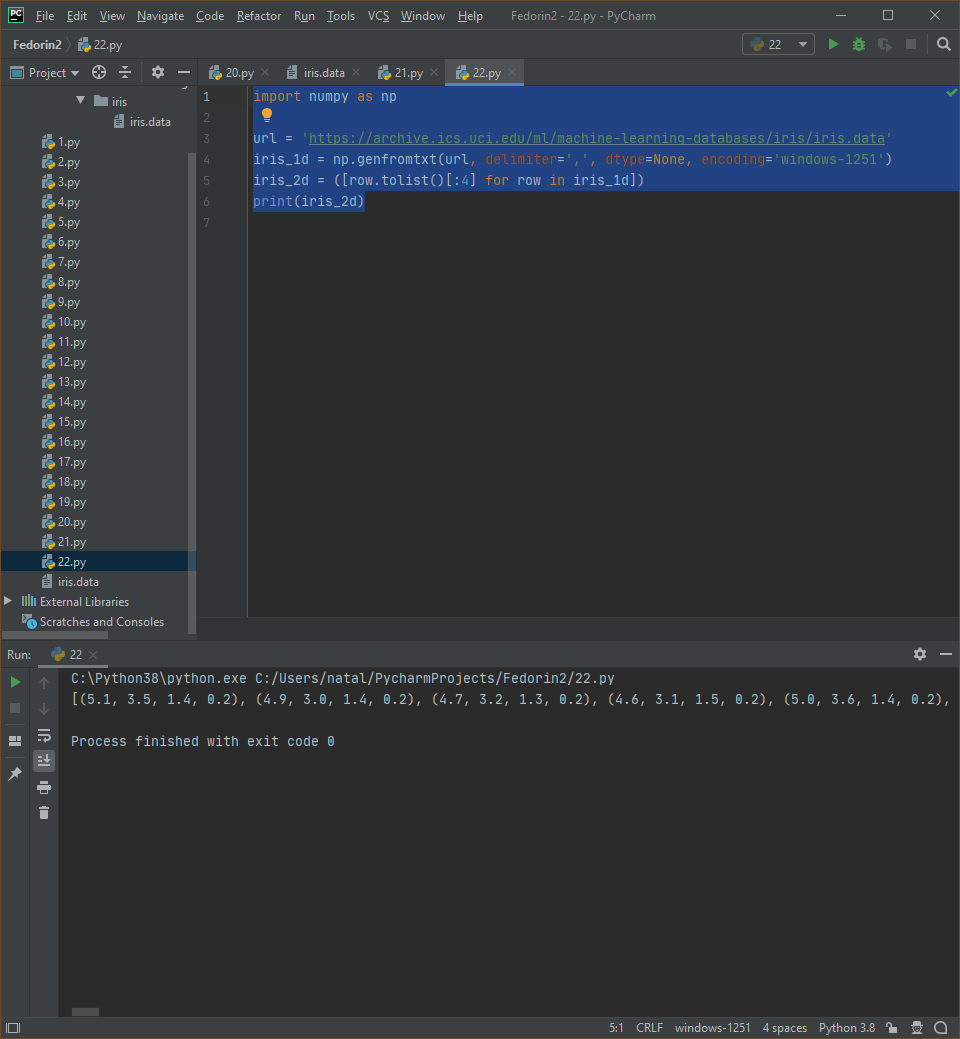


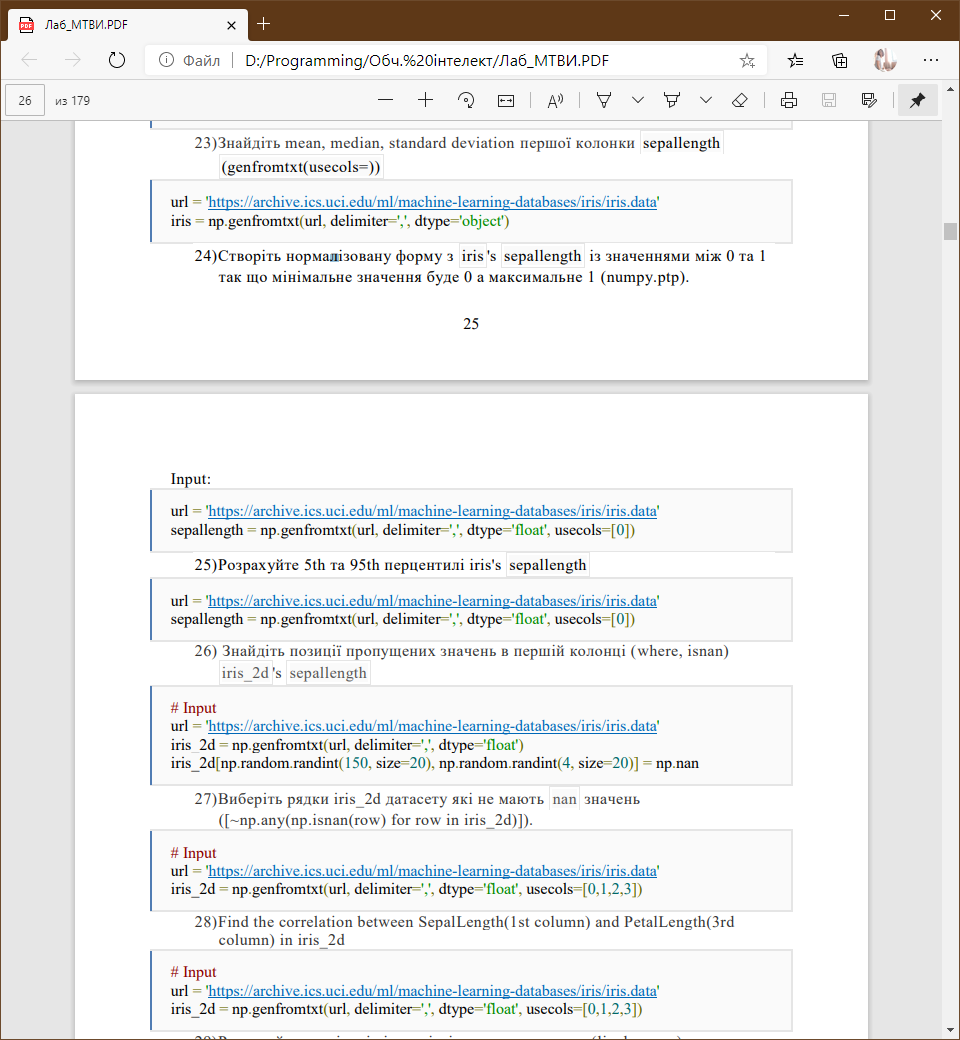


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None, encoding='windows-1251')  
iris\_2d = ([row.tolist()[:4] for row in iris\_1d])  
print(iris\_2d)

**Результат:**

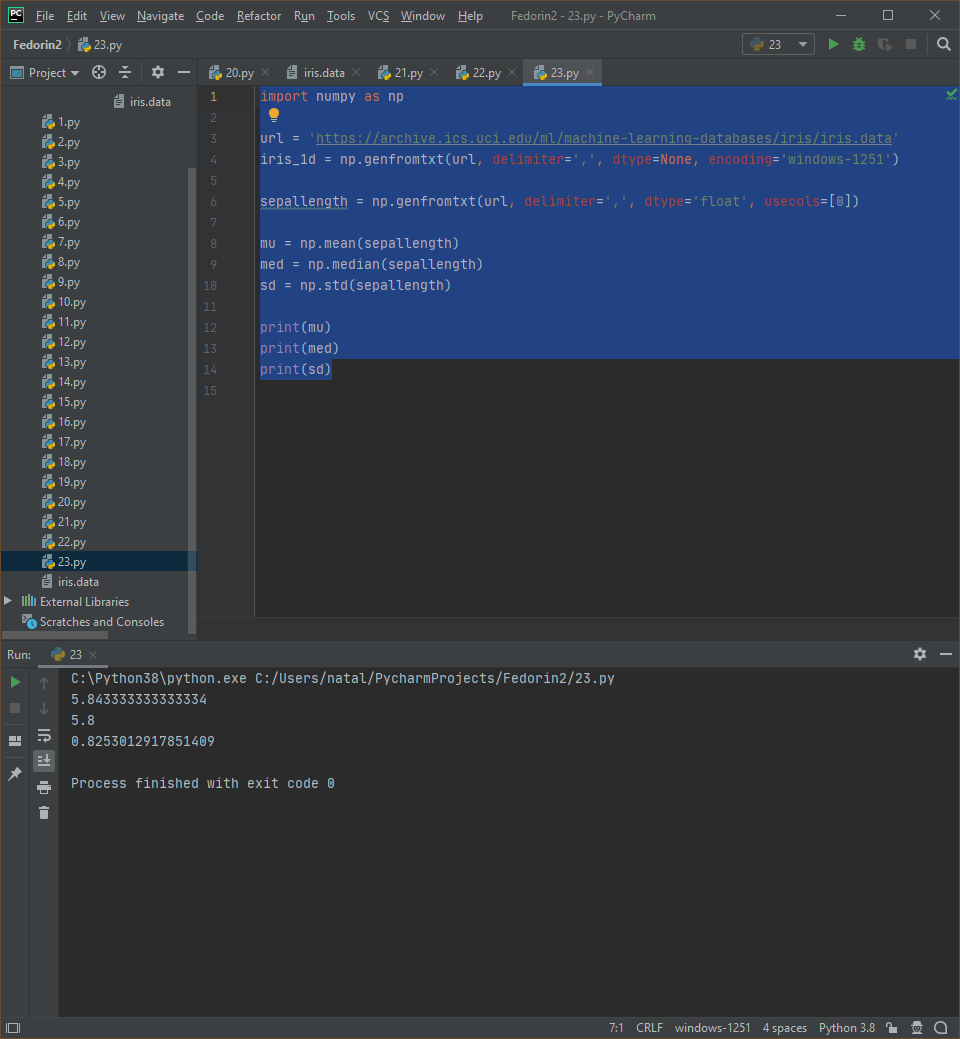


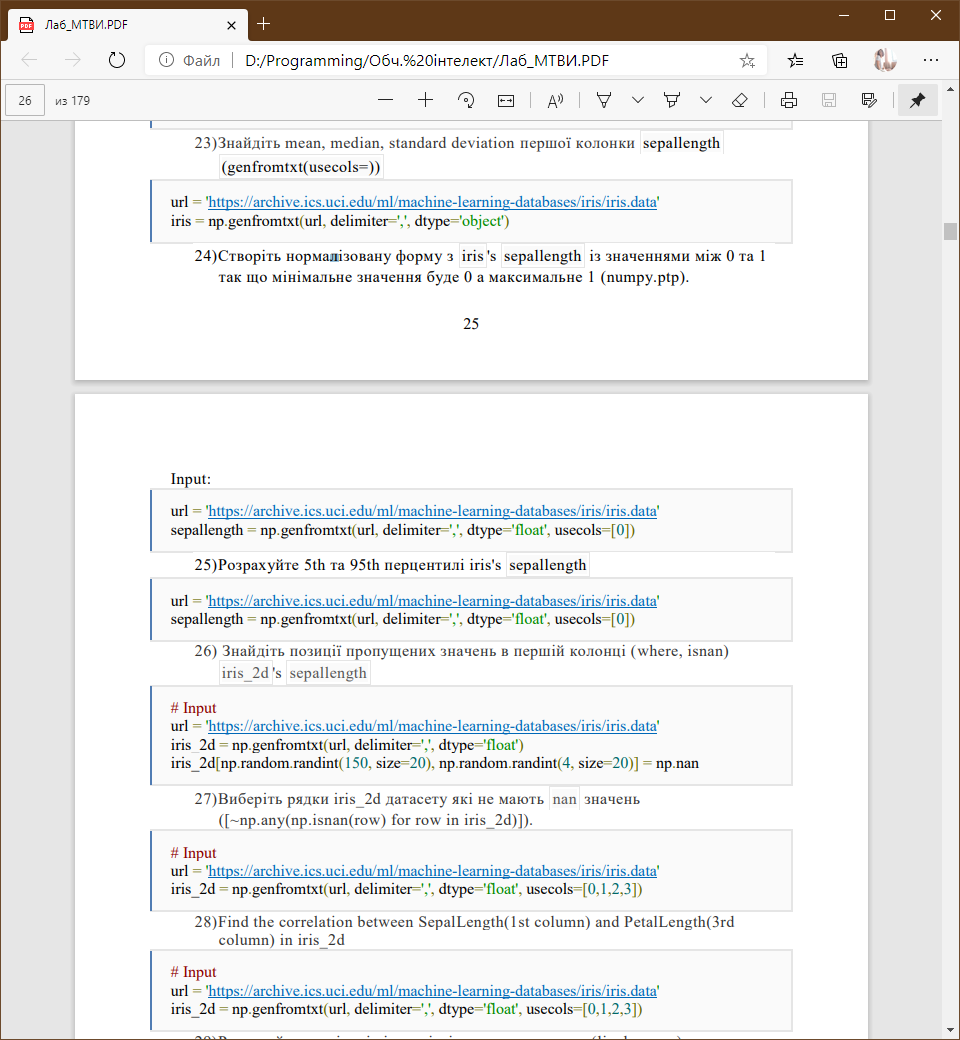


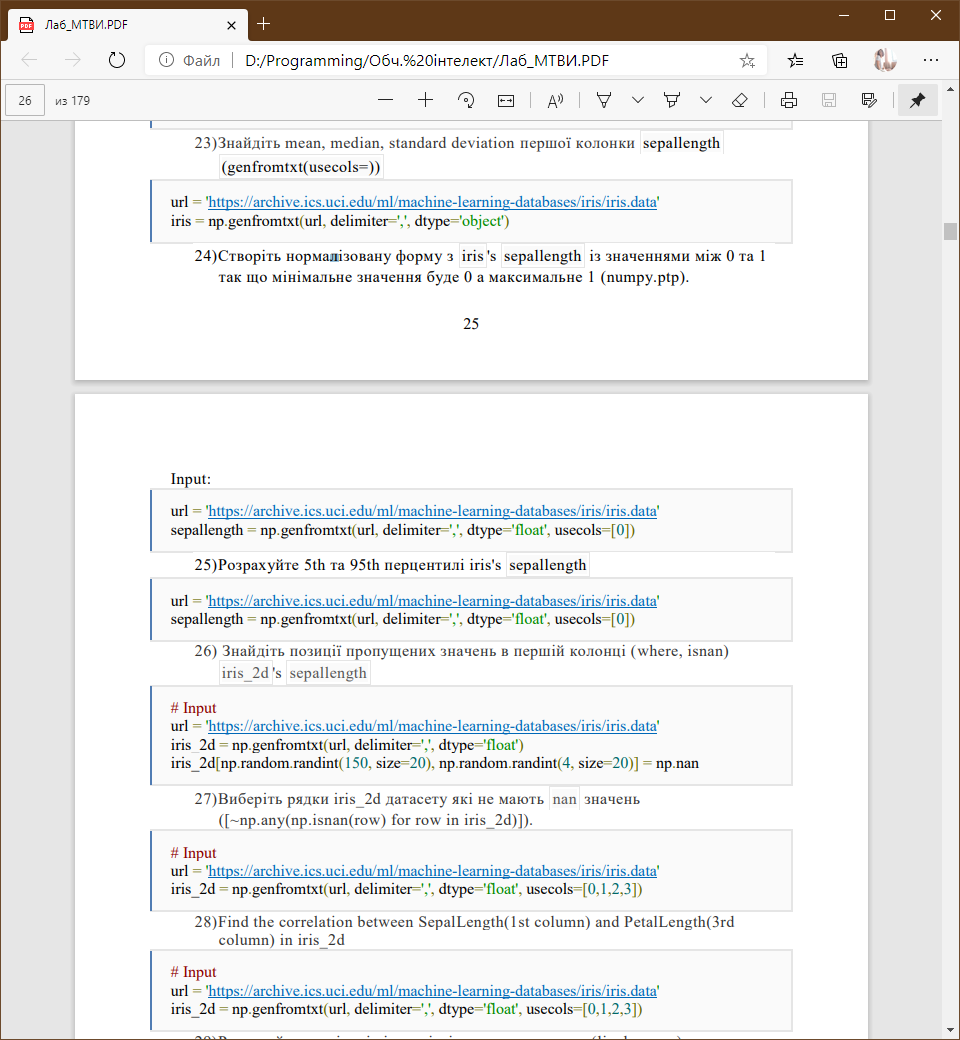
**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None, encoding='windows-1251')  
  
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])  
  
mu = np.mean(sepallength)  
med = np.median(sepallength)  
sd = np.std(sepallength)  
  
print(mu)  
print(med)  
print(sd)

**Результат:**



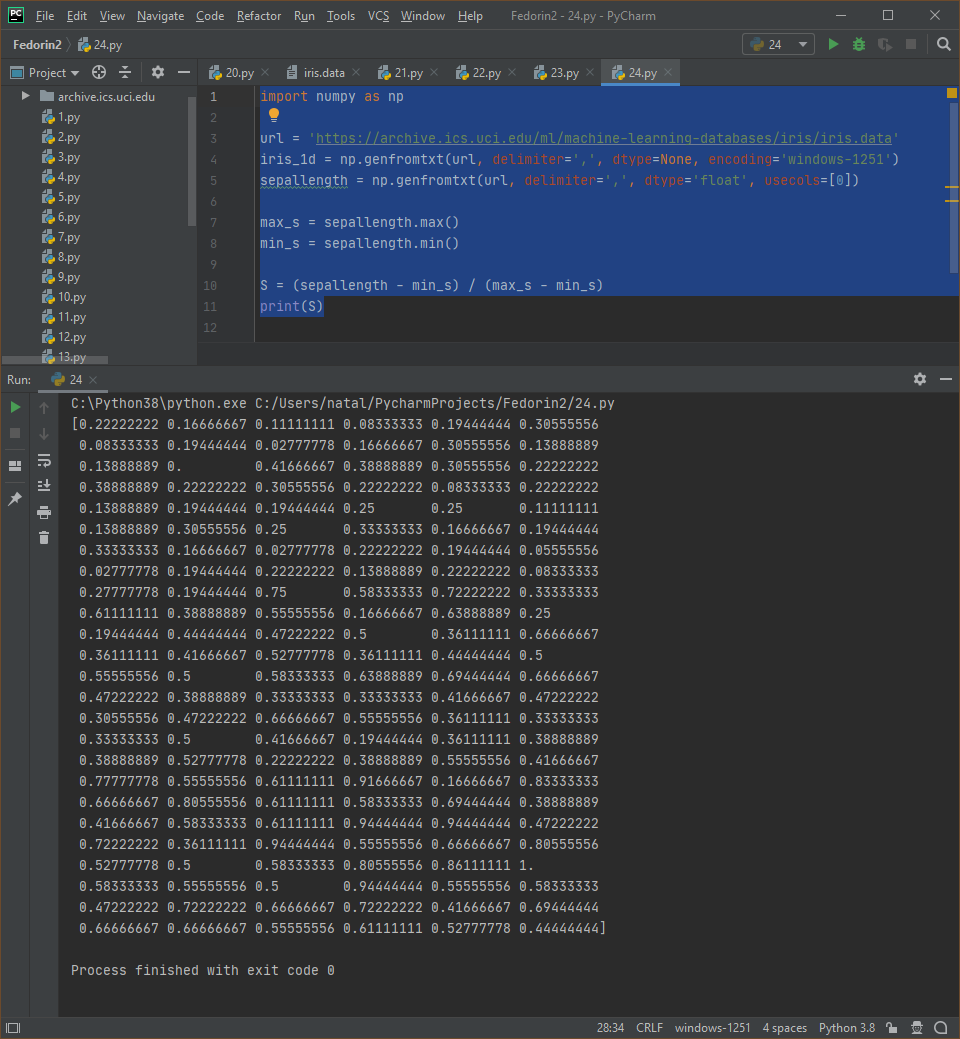


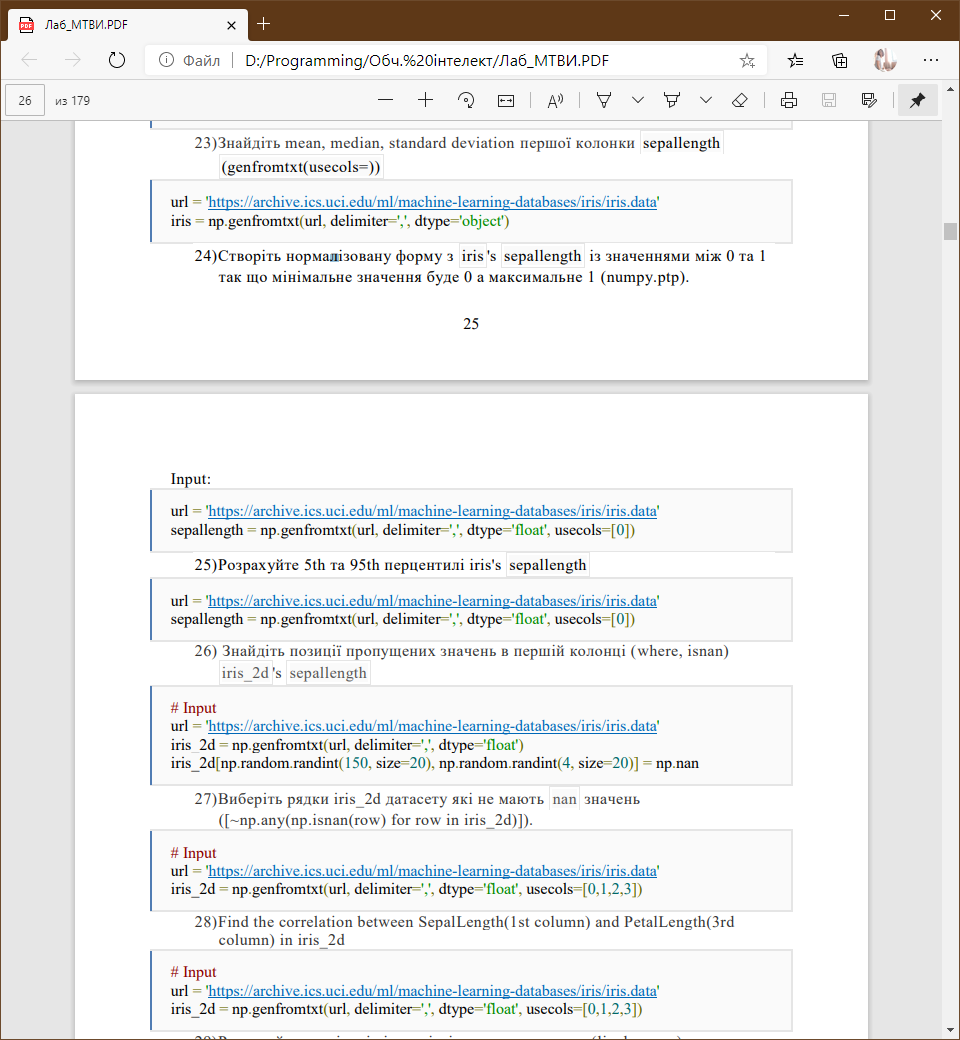


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None, encoding='windows-1251')  
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])  
  
max\_s = sepallength.max()  
min\_s = sepallength.min()  
  
S = (sepallength - min\_s) / (max\_s - min\_s)  
print(S)

**Результат:**

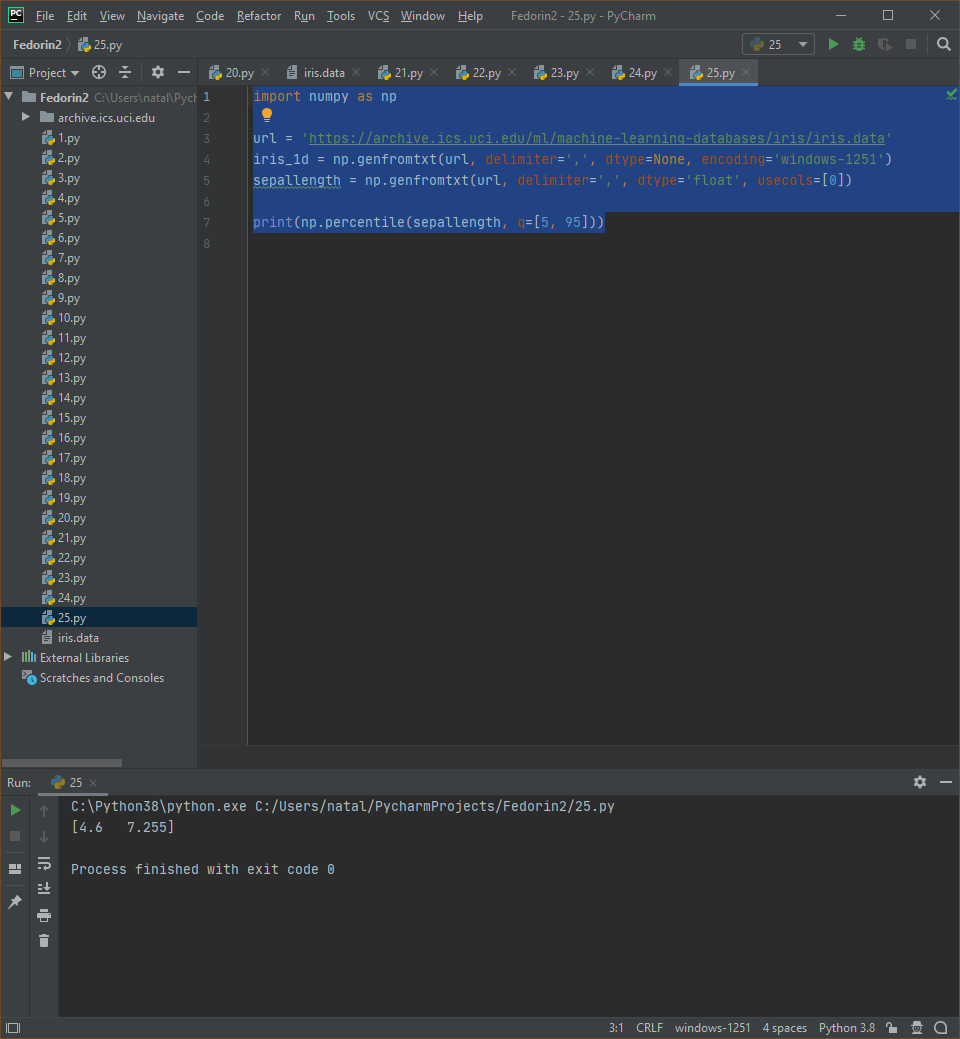


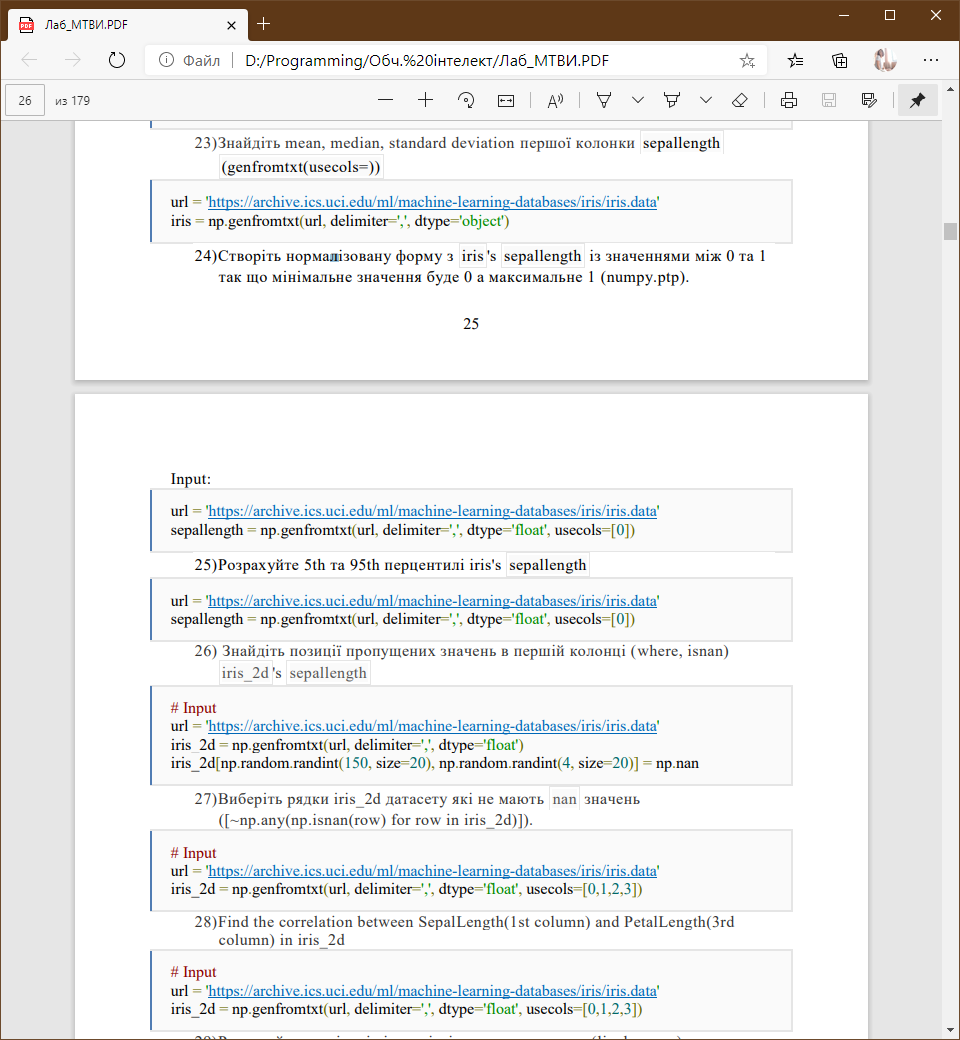


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_1d = np.genfromtxt(url, delimiter=',', dtype=None, encoding='windows-1251')  
sepallength = np.genfromtxt(url, delimiter=',', dtype='float', usecols=[0])  
  
print(np.percentile(sepallength, q=[5, 95]))

**Результат:**

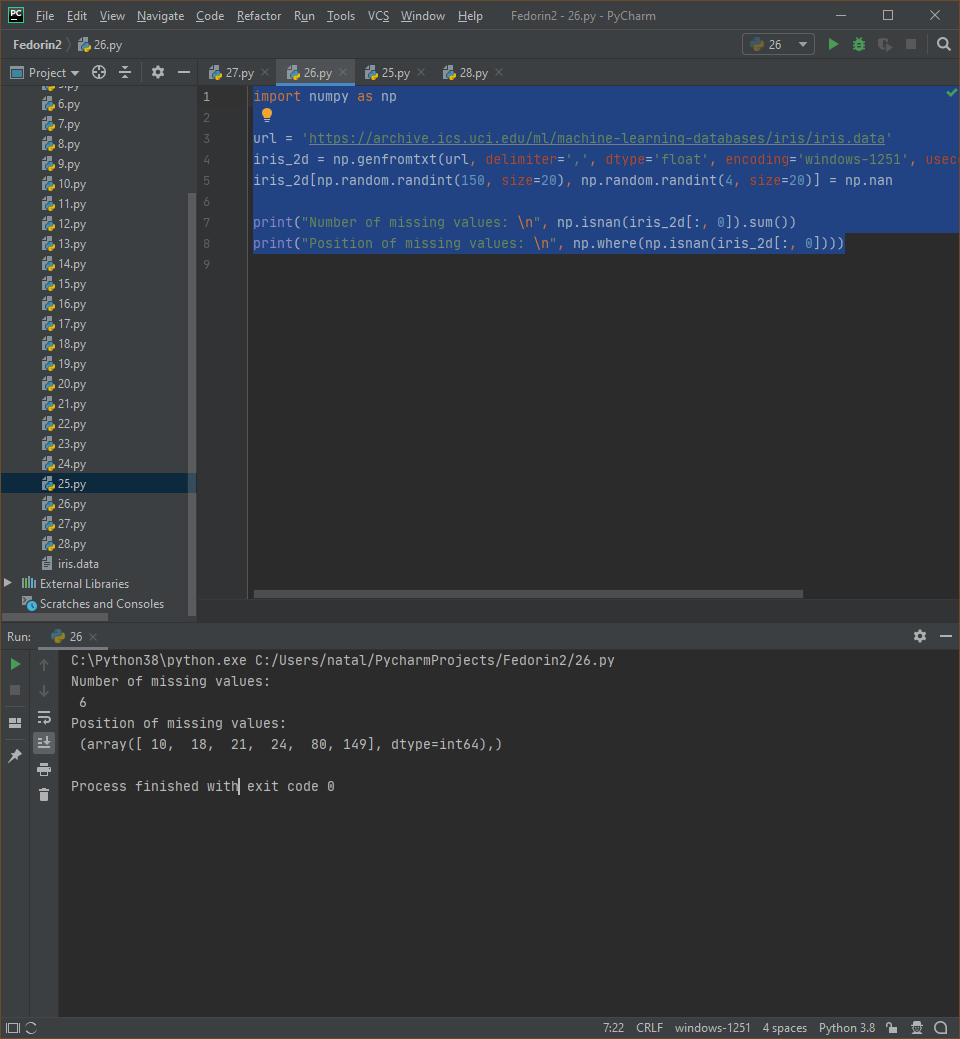


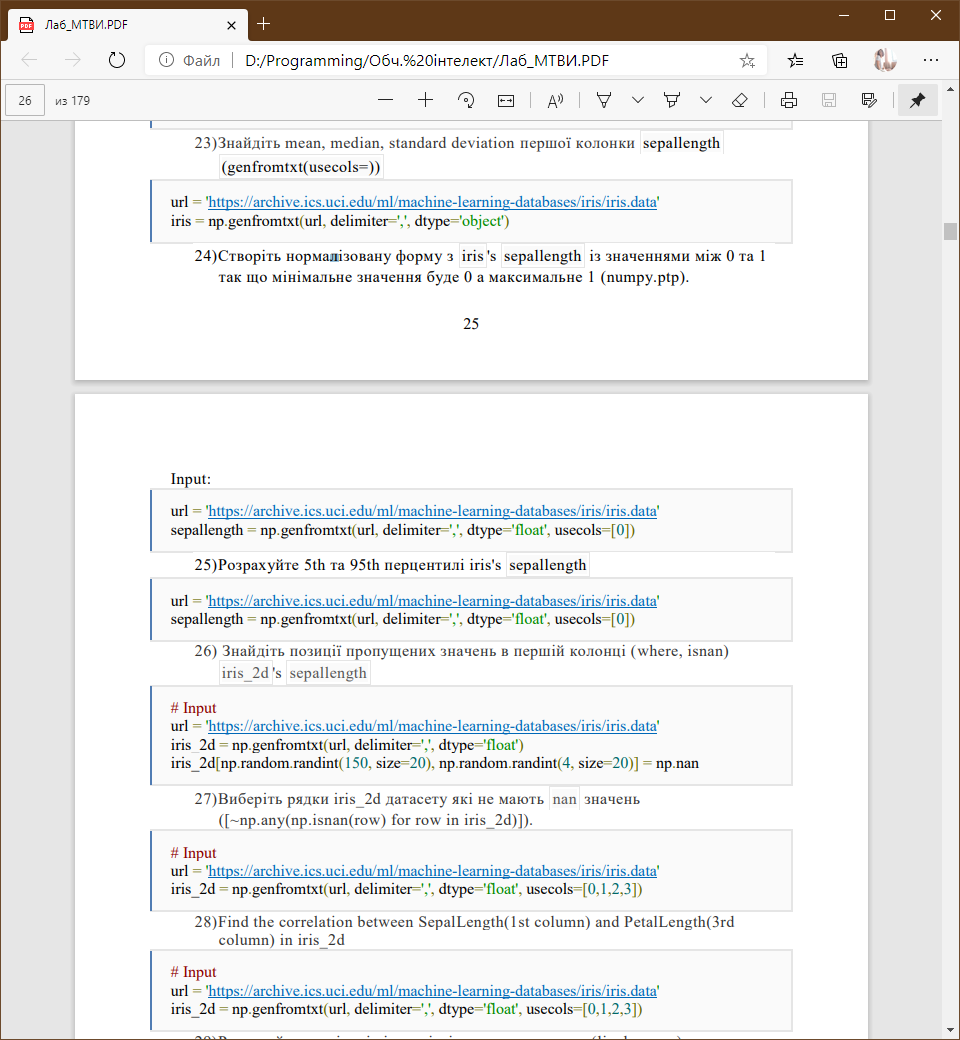


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', encoding='windows-1251', usecols=[0, 1, 2, 3])  
iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan  
  
print("Number of missing values: \n", np.isnan(iris\_2d[:, 0]).sum())  
print("Position of missing values: \n", np.where(np.isnan(iris\_2d[:, 0])))

**Результат:**

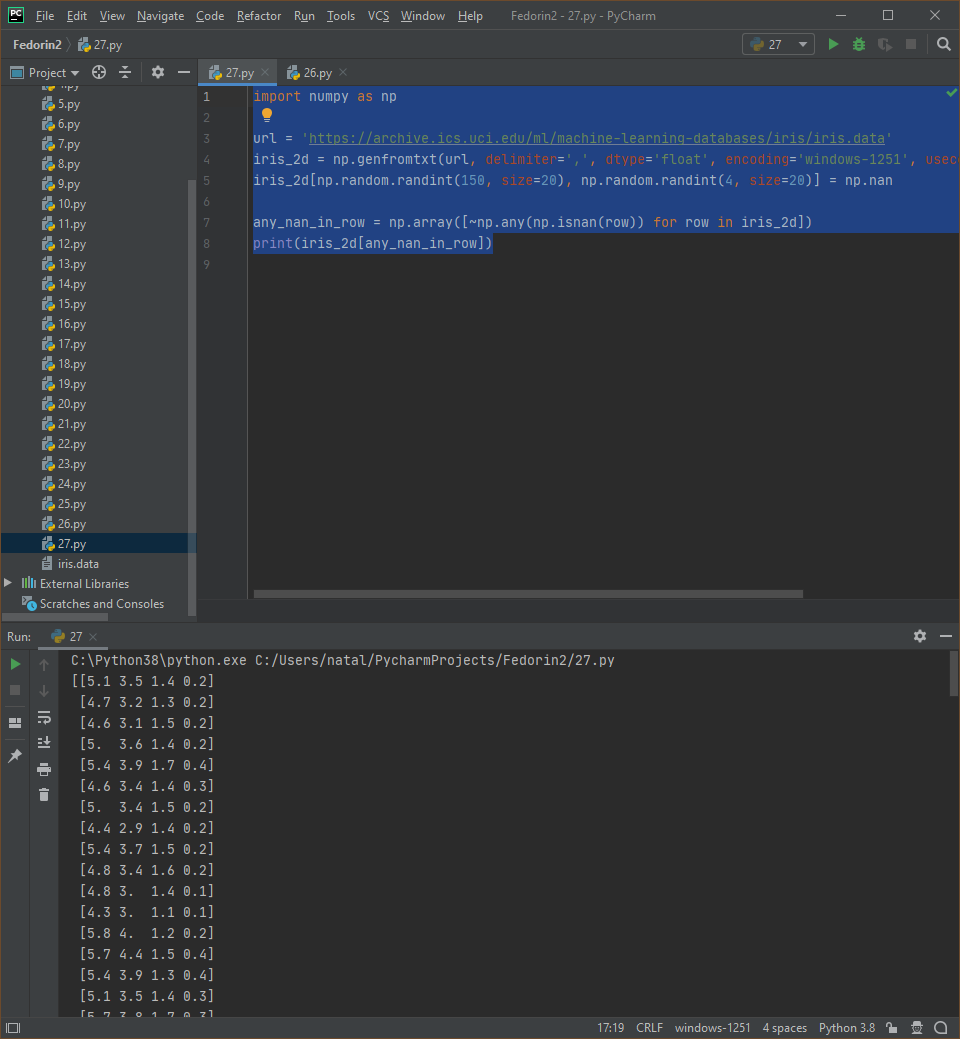


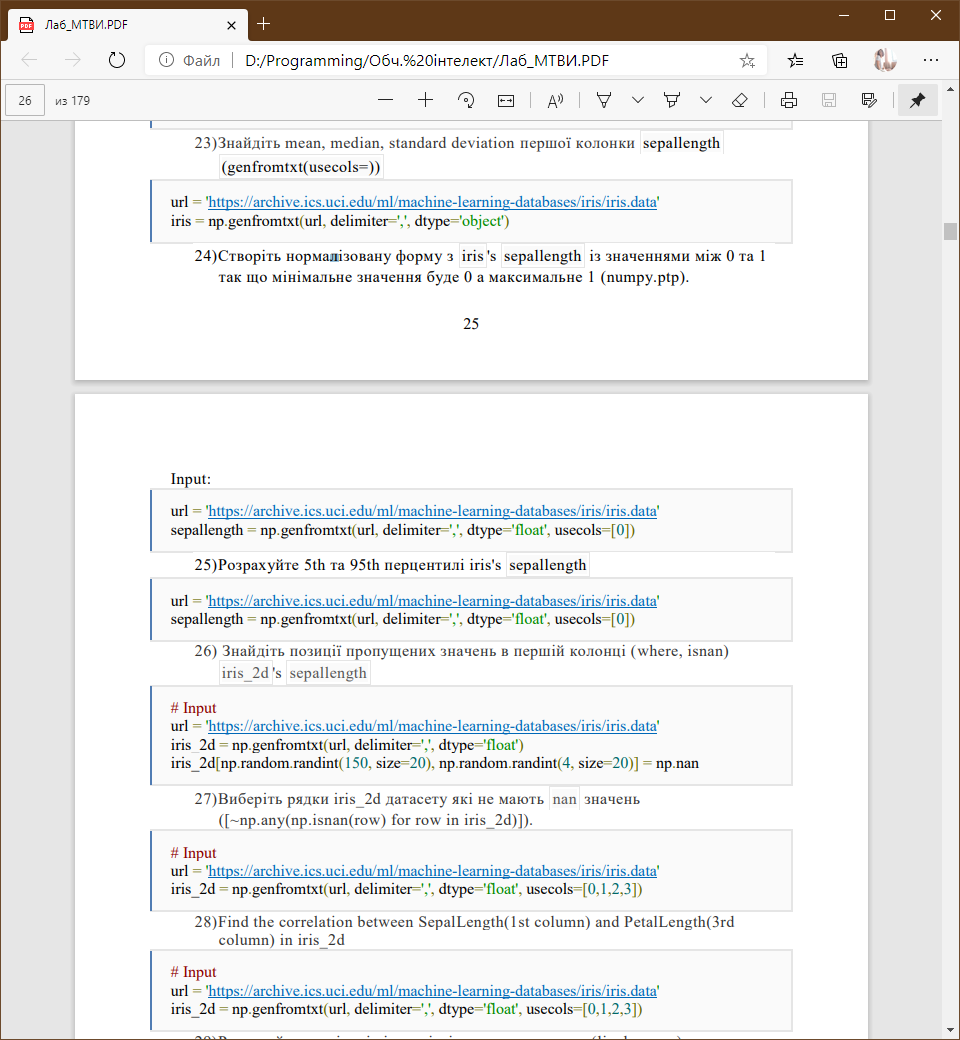


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', encoding='windows-1251', usecols=[0, 1, 2, 3])  
iris\_2d[np.random.randint(150, size=20), np.random.randint(4, size=20)] = np.nan  
  
any\_nan\_in\_row = np.array([~np.any(np.isnan(row)) for row in iris\_2d])  
print(iris\_2d[any\_nan\_in\_row])

**Результат:**

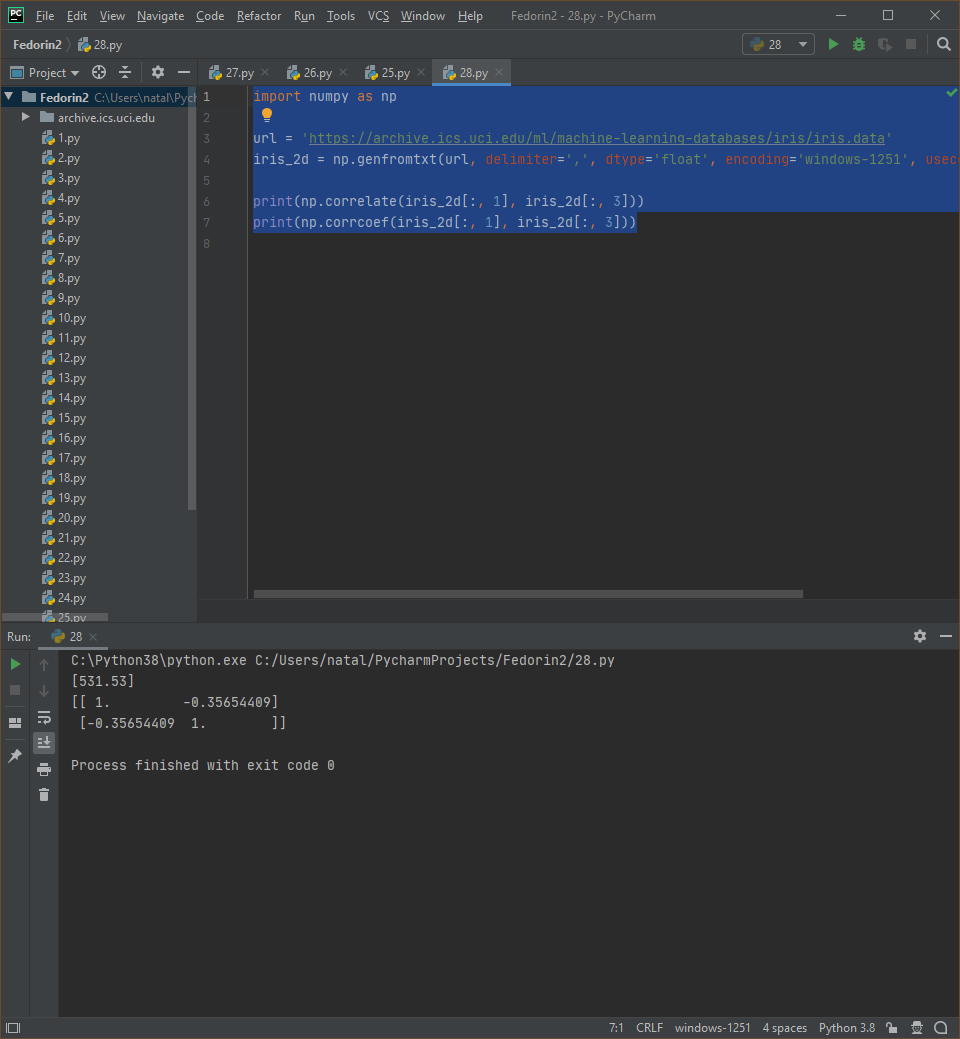


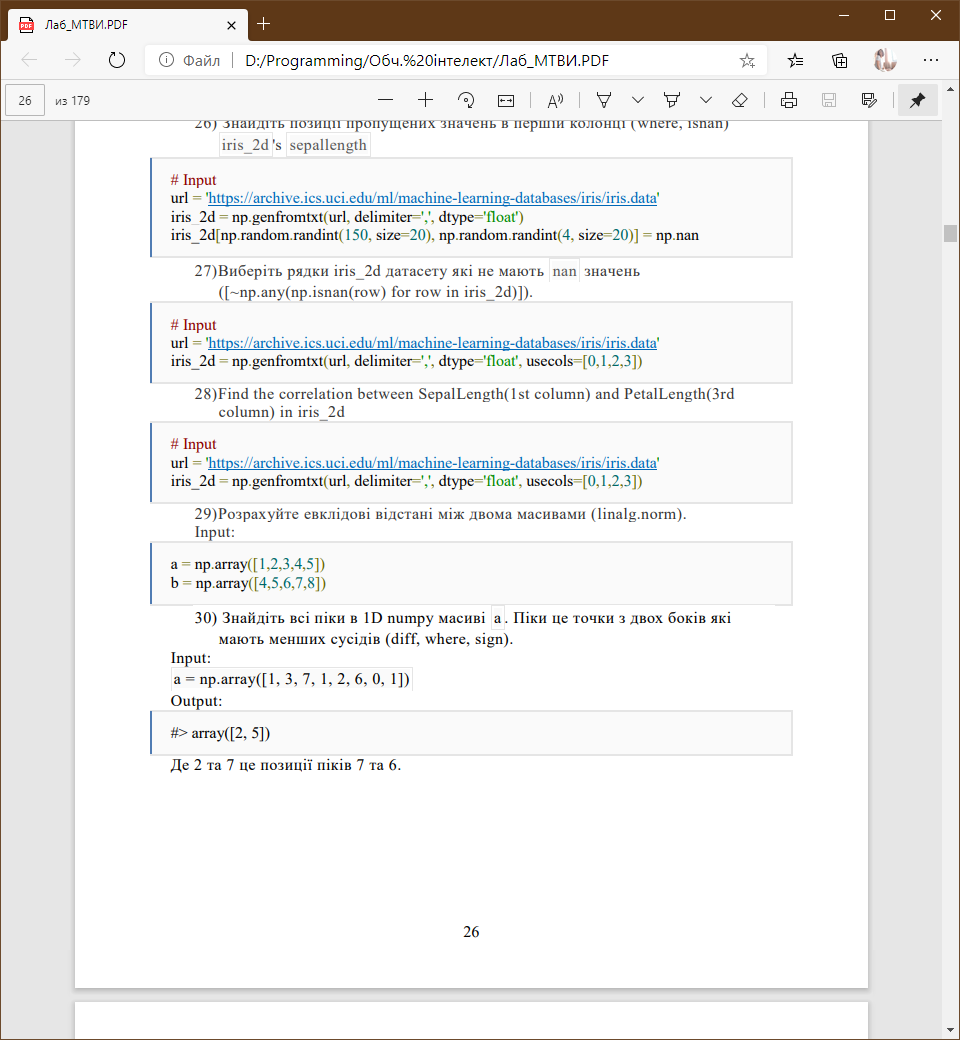


**Програмна реалізація:**

import numpy as np  
  
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'  
iris\_2d = np.genfromtxt(url, delimiter=',', dtype='float', encoding='windows-1251', usecols=[0, 1, 2, 3])  
  
print(np.correlate(iris\_2d[:, 1], iris\_2d[:, 3]))  
print(np.corrcoef(iris\_2d[:, 1], iris\_2d[:, 3]))

**Результат:**

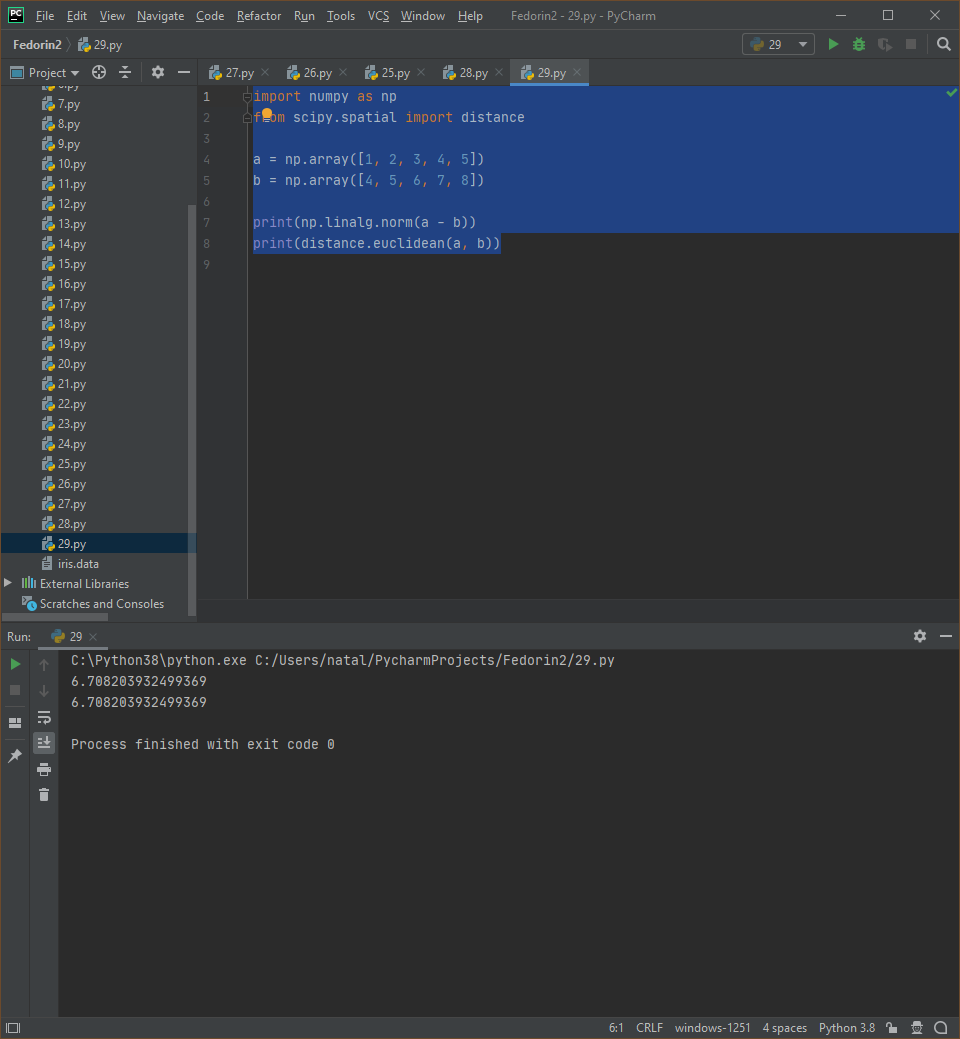


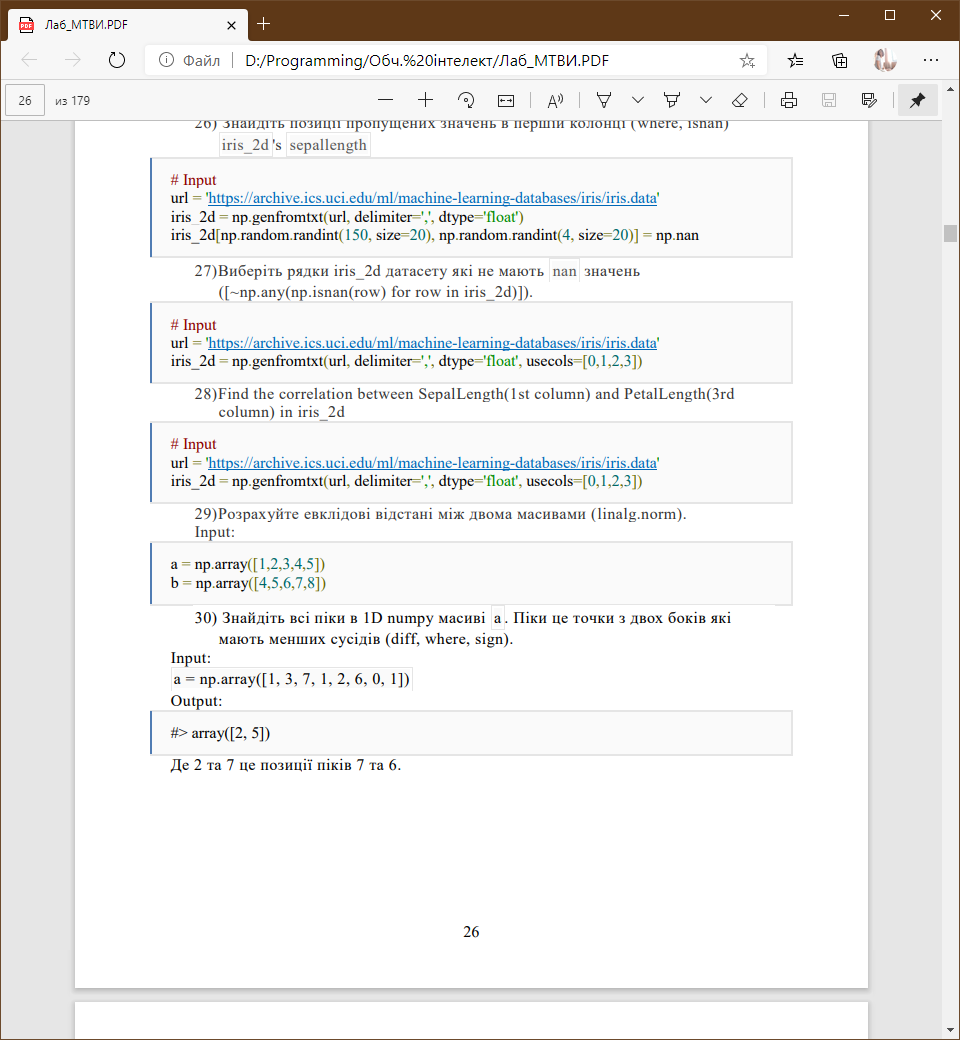


**Програмна реалізація:**

import numpy as np  
from scipy.spatial import distance  
  
a = np.array([1, 2, 3, 4, 5])  
b = np.array([4, 5, 6, 7, 8])  
  
print(np.linalg.norm(a - b))  
print(distance.euclidean(a, b))

**Результат:**





**Програмна реалізація:**

import numpy as np  
  
a = np.array([1, 3, 7, 1, 2, 6, 0, 1])  
  
doublediff = np.diff(np.sign(np.diff(a)))  
print(np.where(doublediff == -2)[0] + 1)

**Результат:**

