Житник Вадим Сергійович №8 ФІТ 3-12

# Знаходження матриці

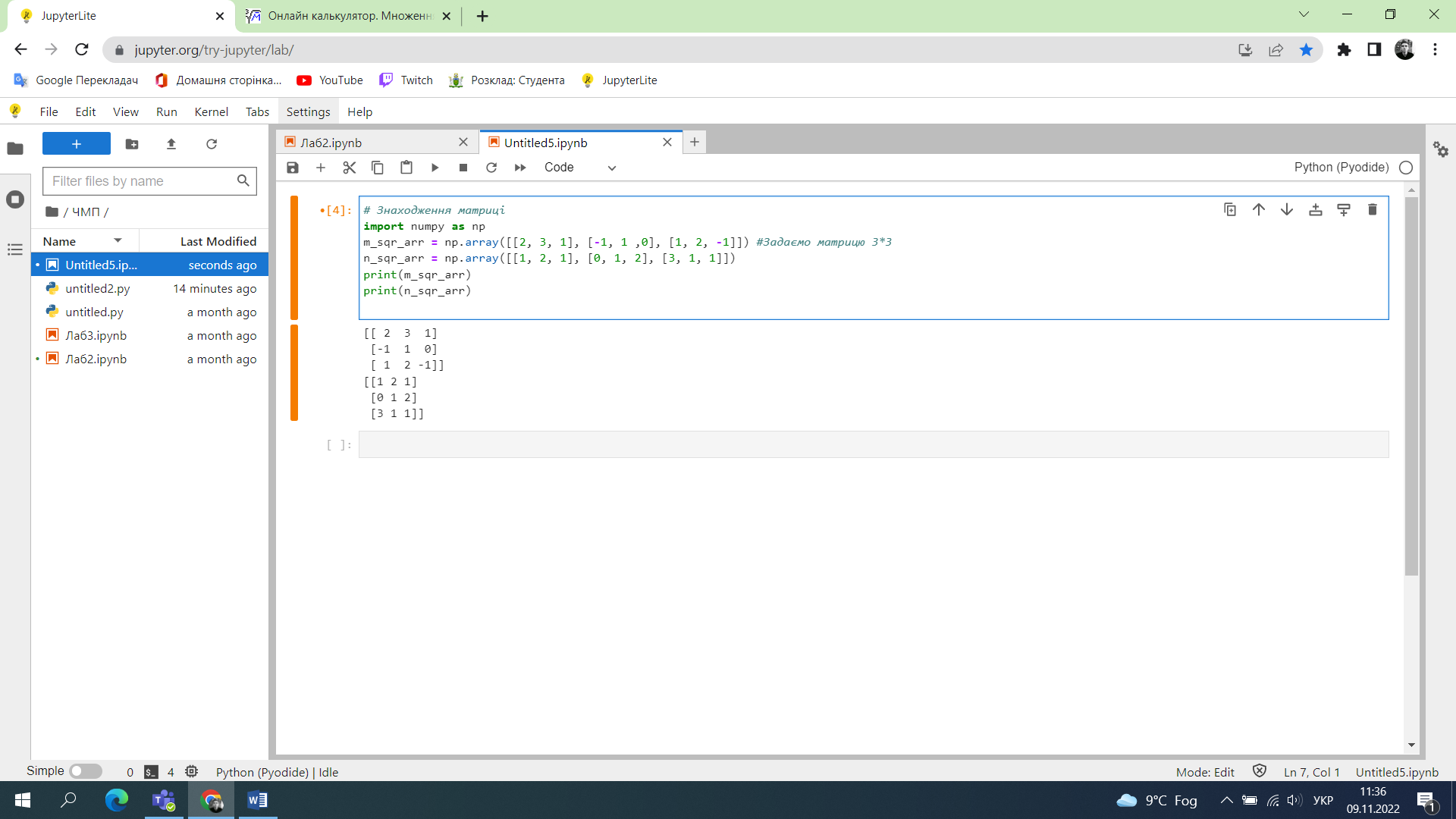
import numpy as np

m\_sqr\_arr = np.array([[2, 3, 1], [-1, 1 ,0], [1, 2, -1]]) #Задаємо матрицю 3\*3

n\_sqr\_arr = np.array([[1, 2, 1], [0, 1, 2], [3, 1, 1]])

print(m\_sqr\_arr)

print(n\_sqr\_arr)



# Піднесення матриці до степеня

a = np.matrix('-1 0 2; 0 1 0; 1 2 -1')

c = 2 \* a

print(c)

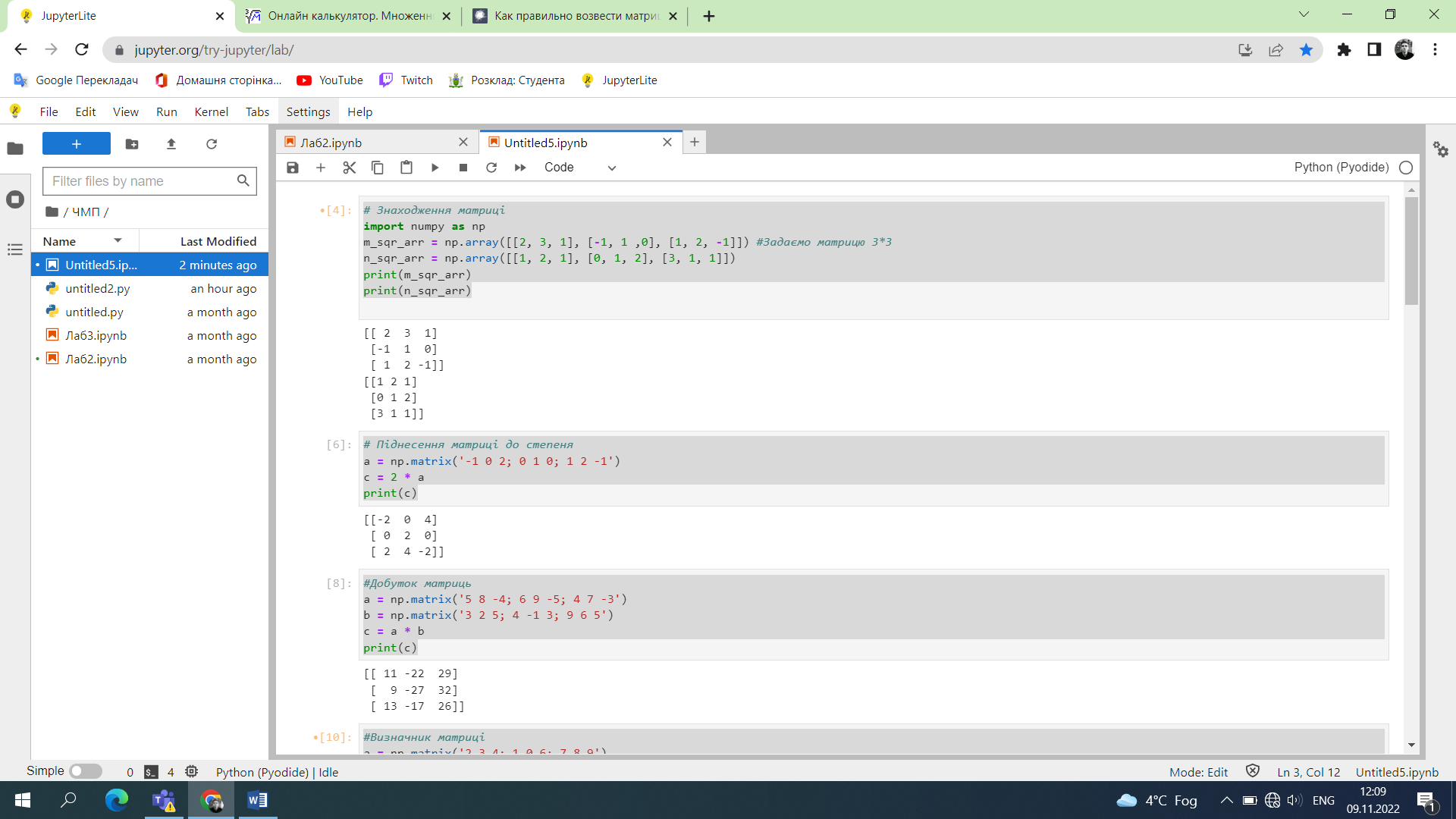
#Добуток матриць

a = np.matrix('5 8 -4; 6 9 -5; 4 7 -3')

b = np.matrix('3 2 5; 4 -1 3; 9 6 5')

c = a \* b

print(c)



#Визначник матриці

a = np.matrix('2 3 4; 1 0 6; 7 8 9')

a\_det = np.linalg.det(a)

print(format(a\_det, '.9g'))

#Визначник матриці

a = np.matrix('2 3 4 1; 1 2 3 4; 3 4 1 2; 4 1 2 3')

a\_det = np.linalg.det(a)

print(format(a\_det, '.9g'))

#Транспонована матриця

a = np.matrix('1 2 -3; 0 1 2; 0 0 1')

print('A=',a)

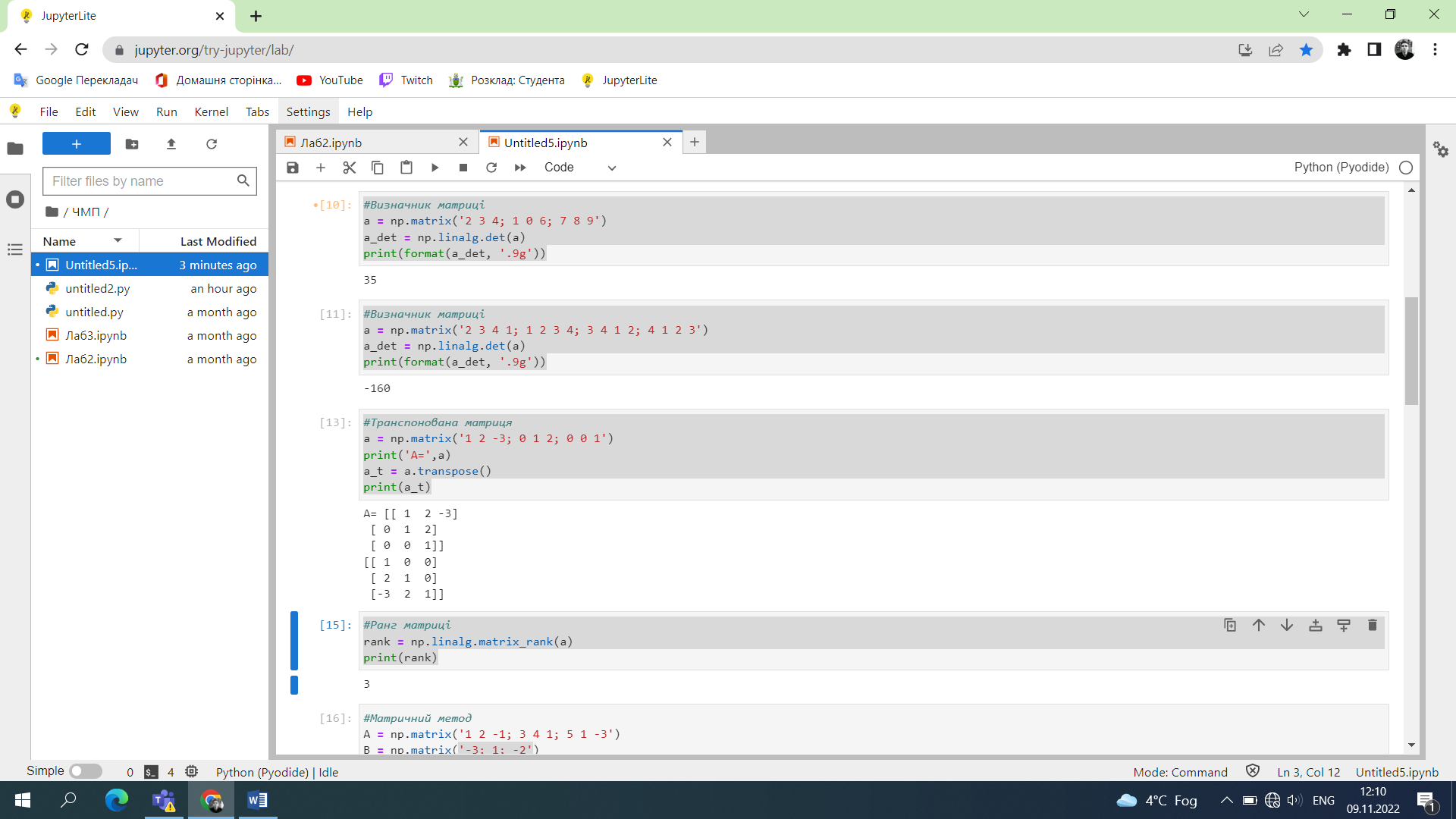
a\_t = a.transpose()

print(a\_t)

#Ранг матриці

rank = np.linalg.matrix\_rank(a)

print(rank)



#Матричний метод

A = np.matrix('1 2 -1; 3 4 1; 5 1 -3')

B = np.matrix('-3; 1; -2')

print('A=', A)

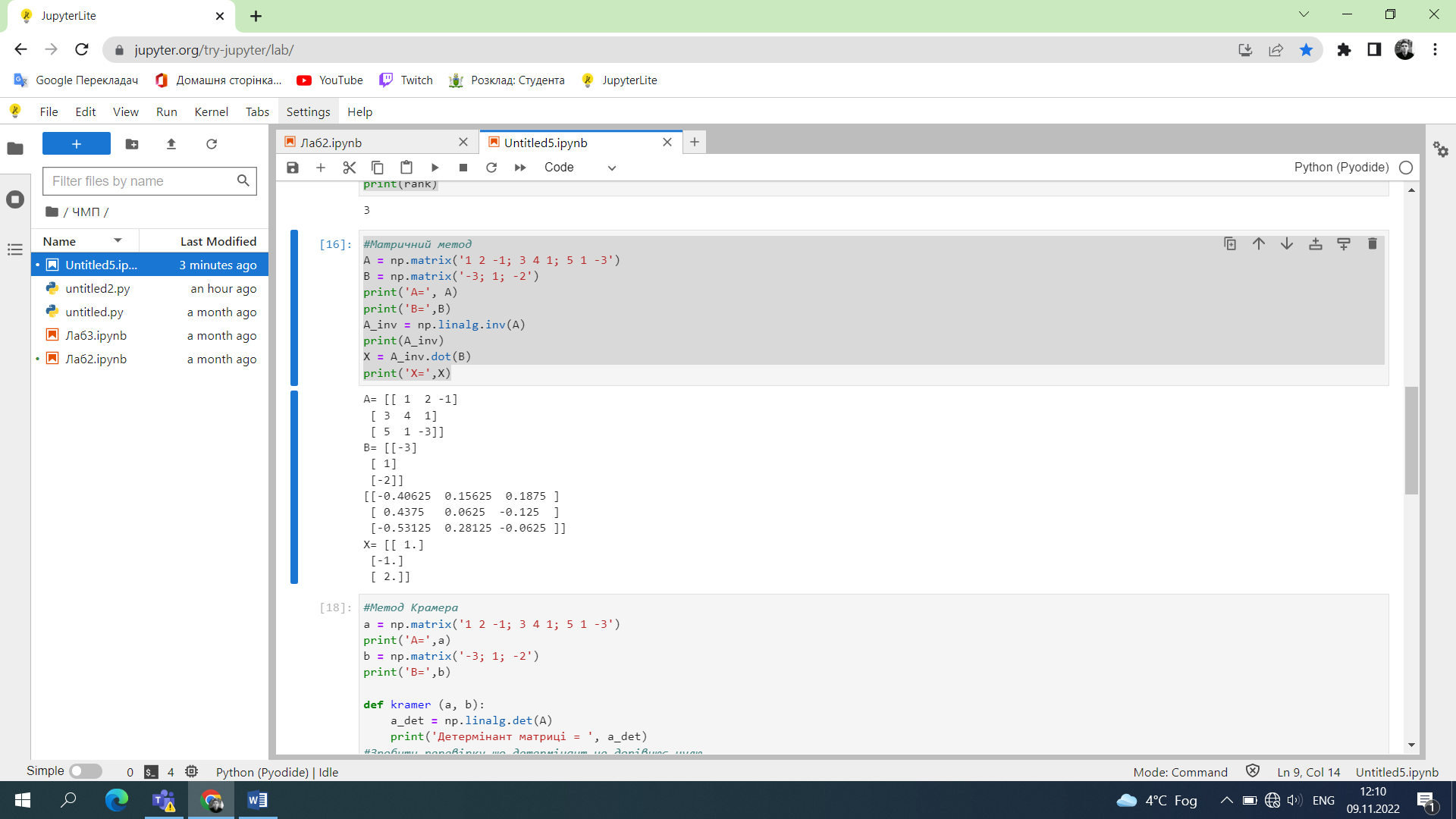
print('B=',B)

A\_inv = np.linalg.inv(A)

print(A\_inv)

X = A\_inv.dot(B)

print('X=',X)



#Метод Крамера

a = np.matrix('1 2 -1; 3 4 1; 5 1 -3')

print('A=',a)

b = np.matrix('-3; 1; -2')

print('B=',b)

def kramer (a, b):

a\_det = np.linalg.det(A)

print('Детермінант матриці = ', a\_det)

#Зробити реревірку,що детермінант не дорівнює нулю

if (a\_det != 0):

print ('Розв\*язуємо систему')

x\_m = np.matrix(a)

x\_m[:, 0] = b # формування допоміжної матриці (1 ст. замінюємо на ст. b)

print('x\_m=', x\_m)

y\_m = np.matrix(a) #2 ст. замінюємо на ст. b

y\_m[:, 1] = b #2 c

print('y\_m=',y\_m)

z\_m = np.matrix(a) #3 ст. замінюємо на ст. b

z\_m[:, 2] = b

print('z\_m=',z\_m)

x = np.linalg.det(x\_m) / a\_det

y = np.linalg.det(y\_m) / a\_det

z = np.linalg.det(z\_m) / a\_det

print('X = ', round(x,5))

print('Y=', round(y,5))

print('Z=', round(z,5))

else:

print('Розв\*язків немає')

kramer(a,b)

