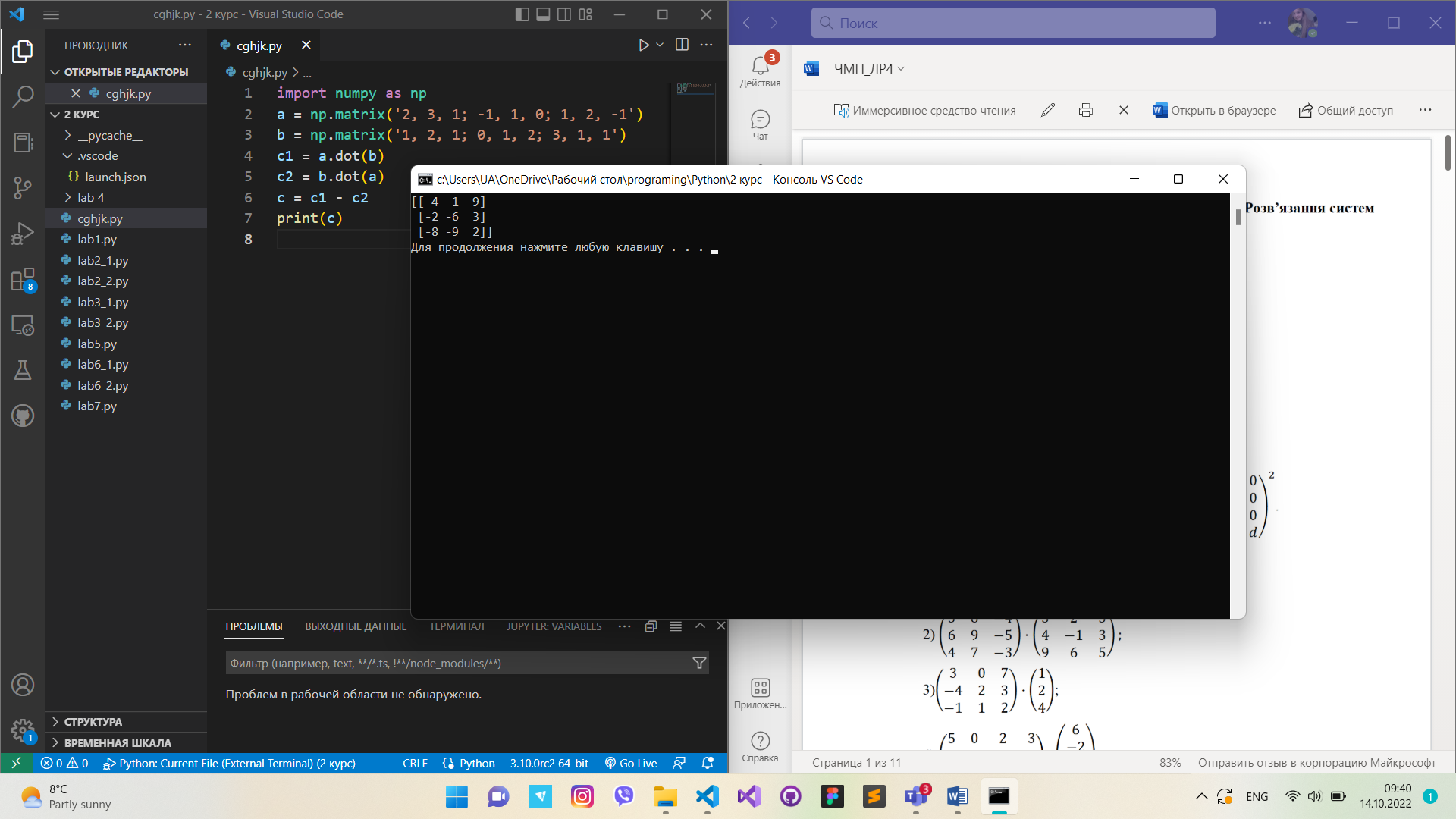
**Лабораторна робота №4**

**Сухина Денис ФІТ 2-6**

**25 варіант**

**1. Знайти матрицю 𝐴𝐵−𝐵𝐴: 2)**

import numpy as np

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

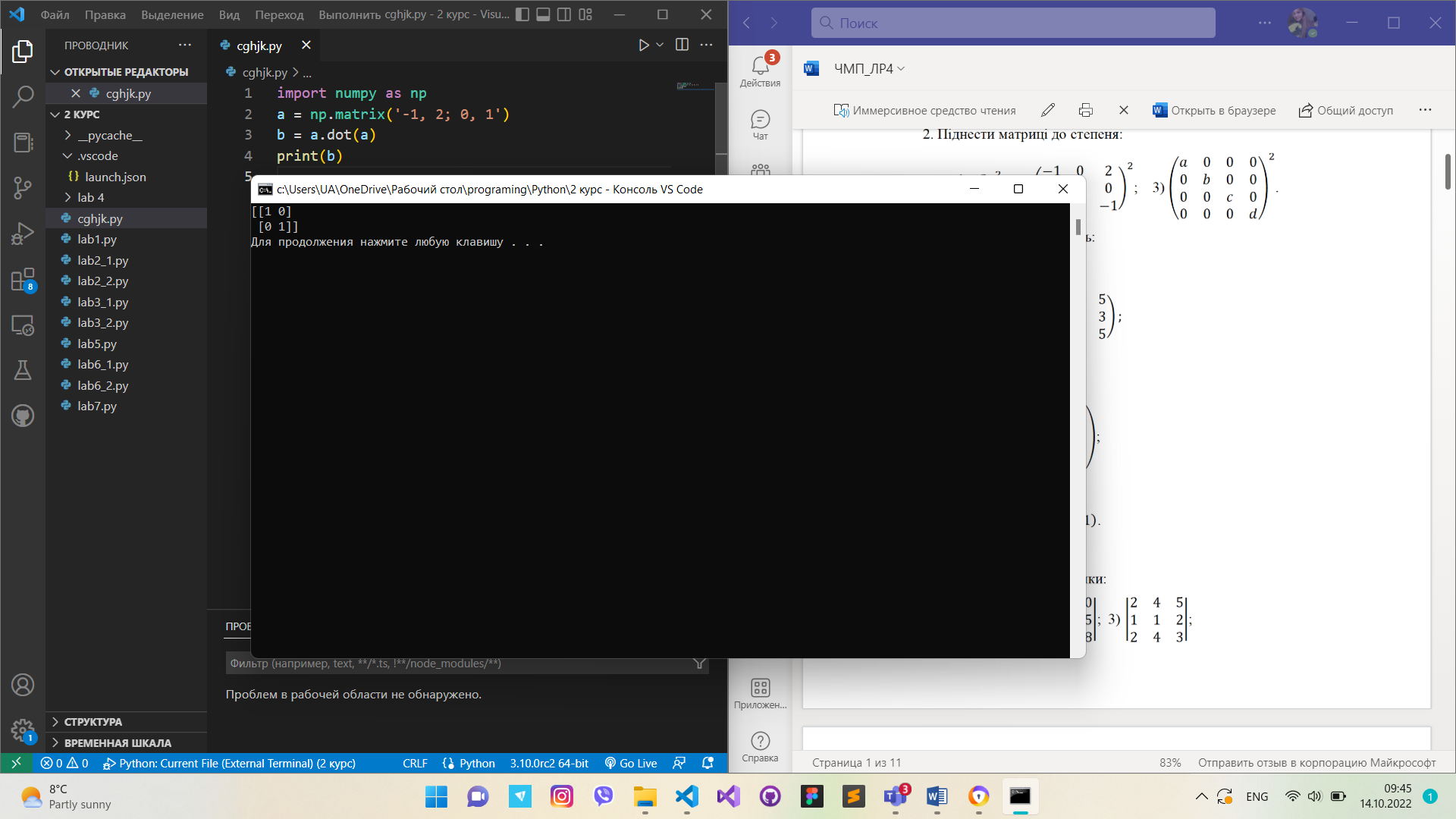
b = np.matrix('1, 2, 1; 0, 1, 2; 3, 1, 1')

c1 = a.dot(b)

c2 = b.dot(a)

c = c1 - c2

print(c))

**2. Піднести матриці до степеня: 1)**

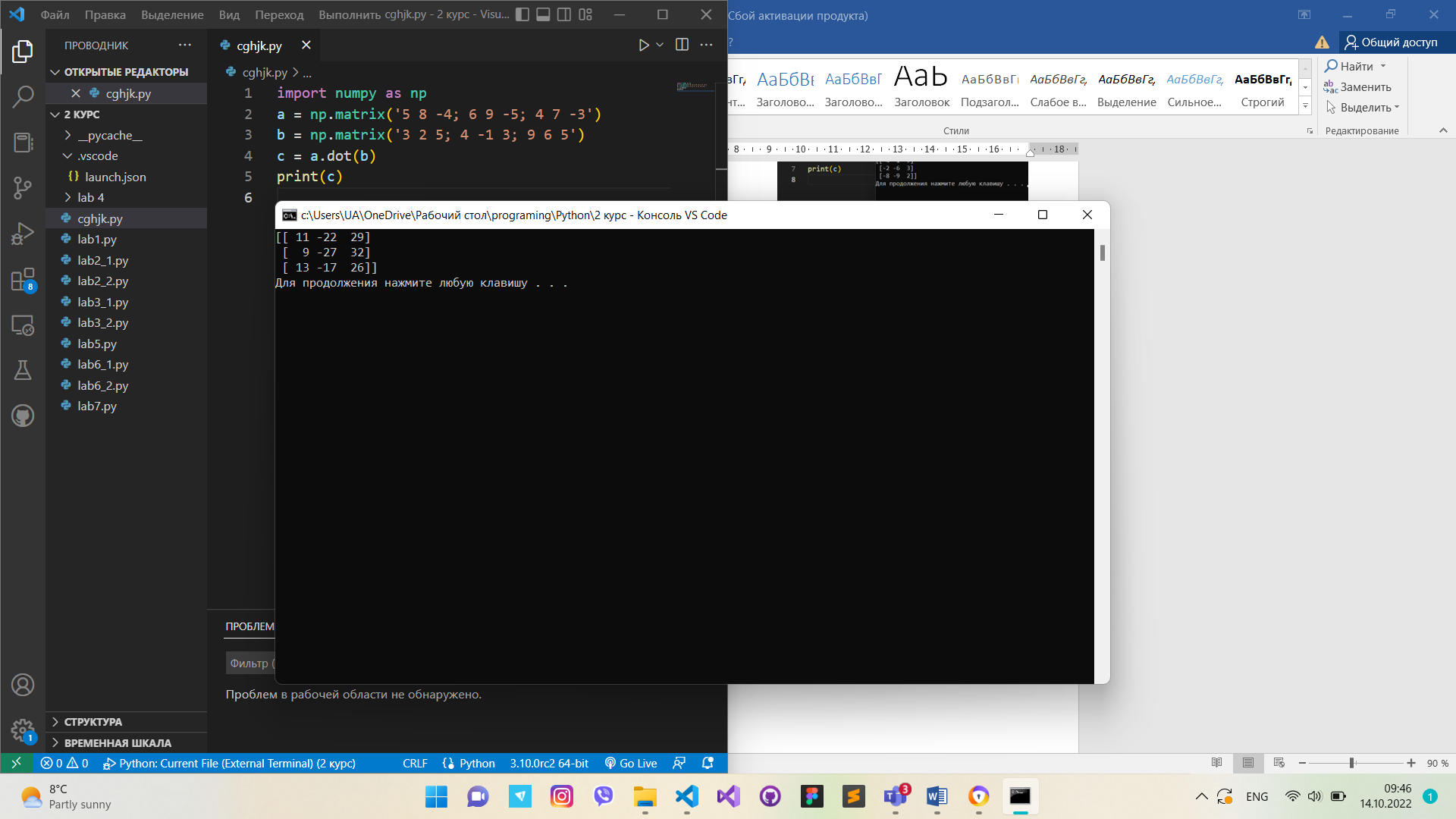
import numpy as np

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

b = a.dot(a)

print(b)

**3. Знайти добуток матриць: 2)**

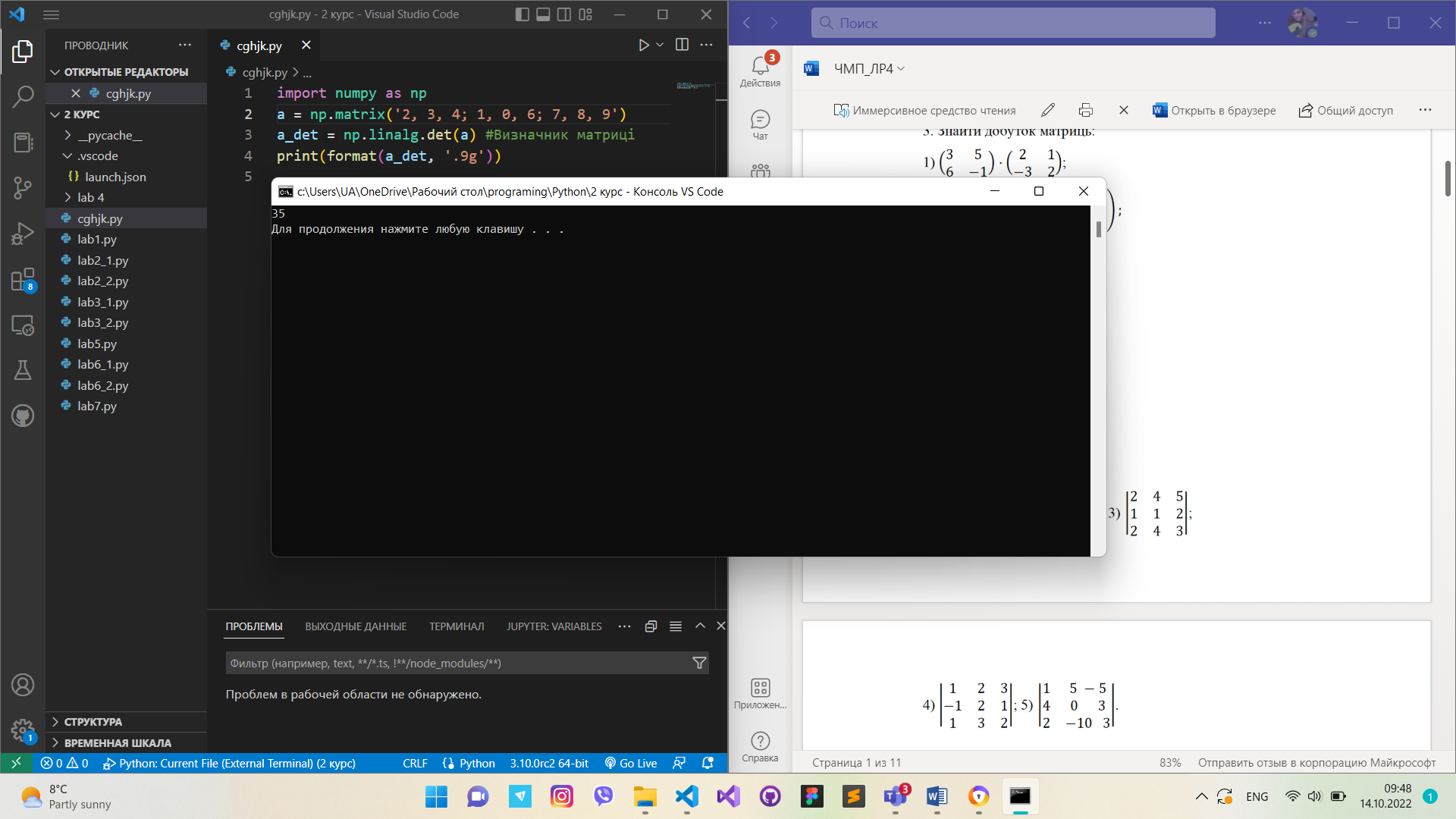
import numpy as np

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.dot(b)

print(c)

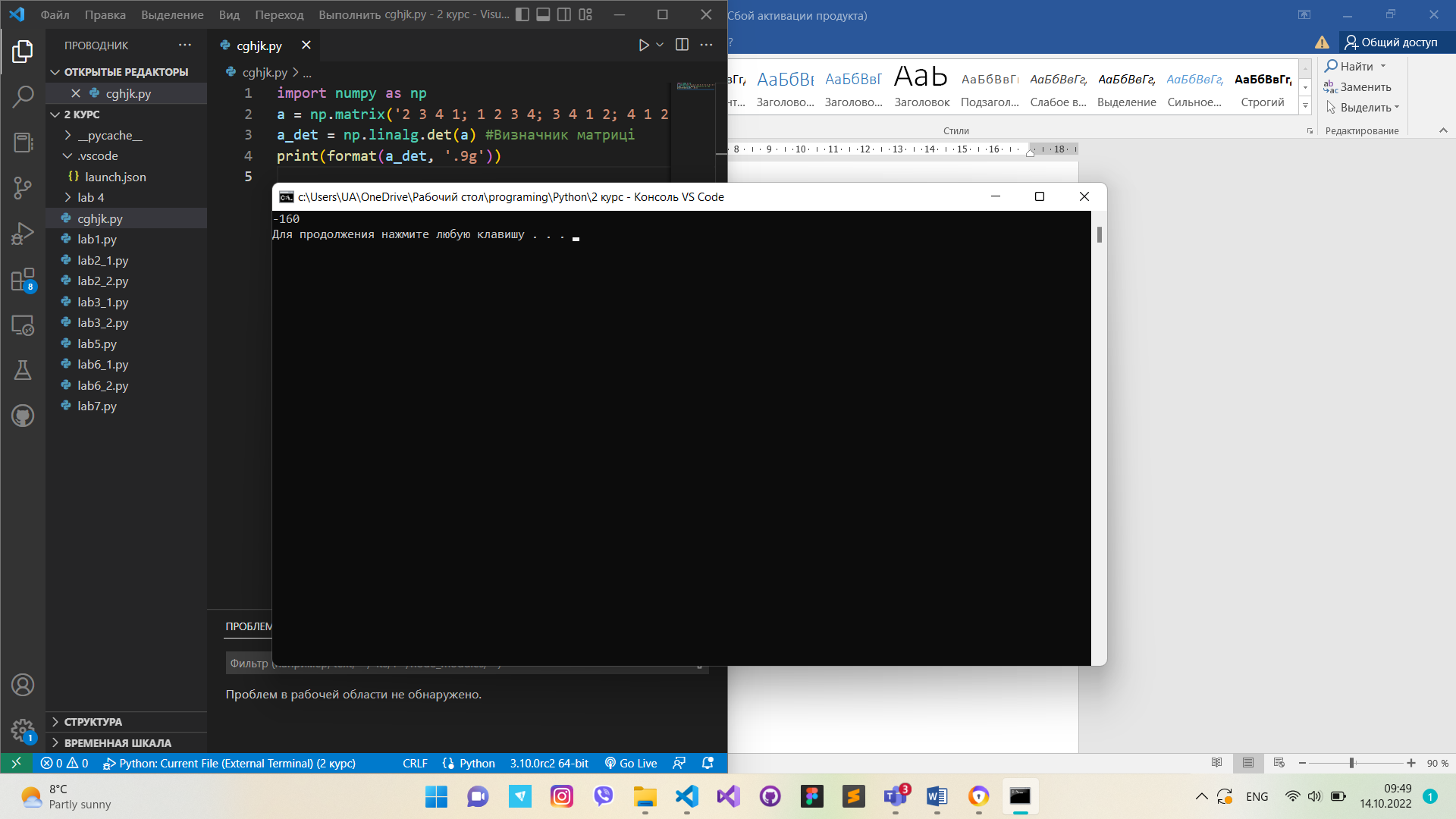
**4. Обчислити визначники: 1)**

import numpy as np

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

a\_det = np.linalg.det(a) #Визначник матриці

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

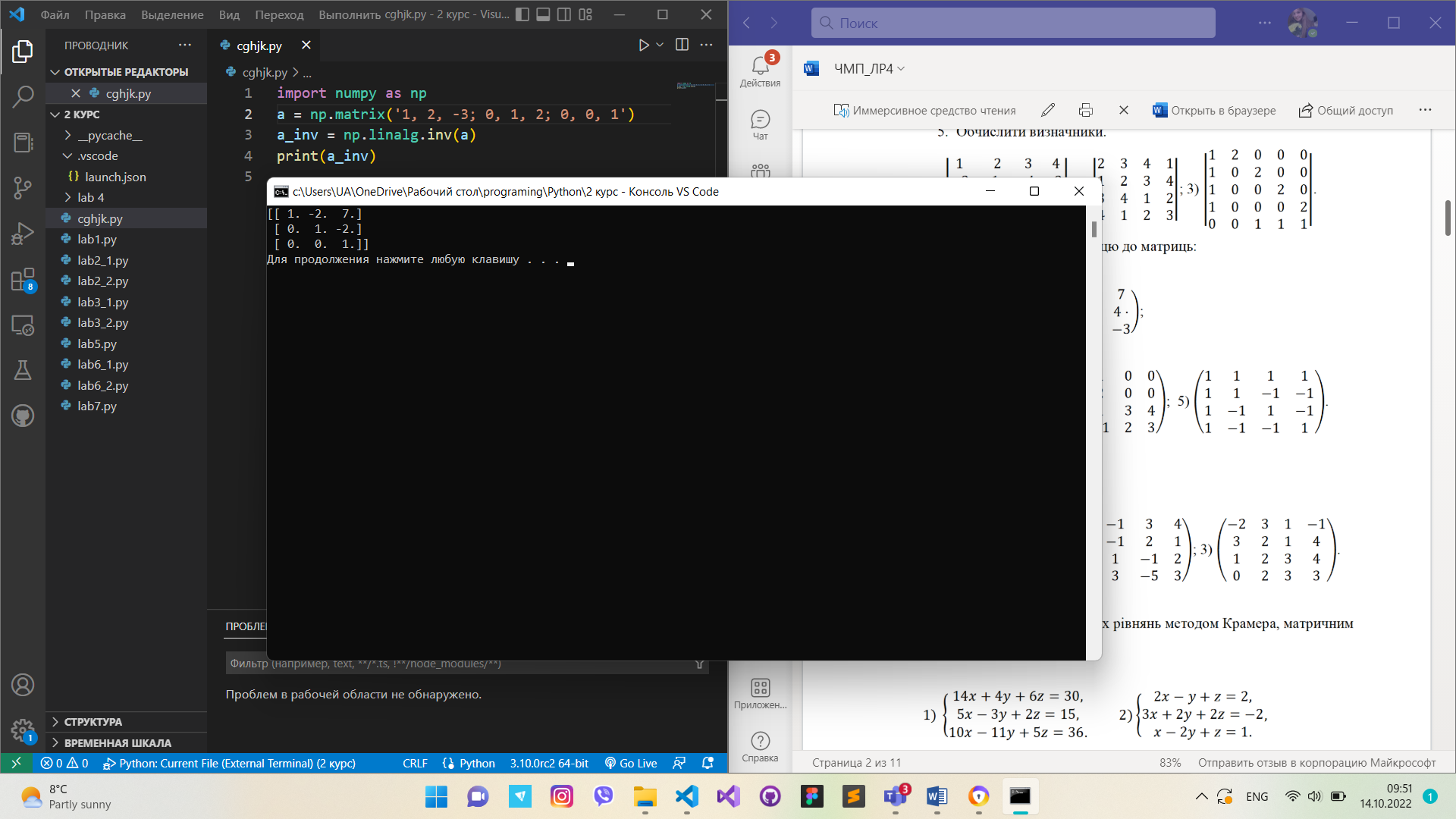
**5. Обчислити визначники. 2)**

import numpy as np

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) #Визначник матриці

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

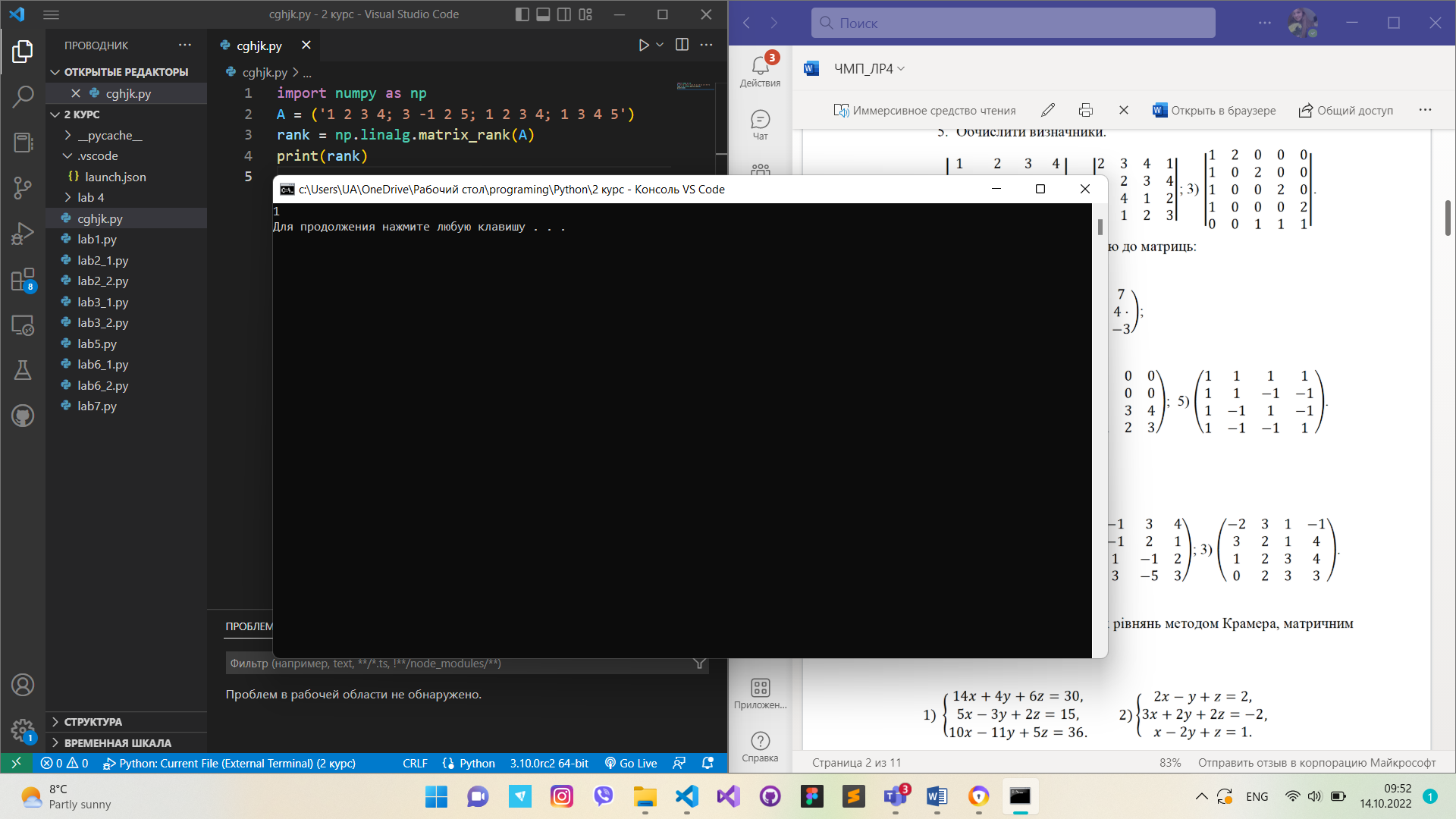
**6. Знайти обернену матрицю до матриць: 1)**

import numpy as np

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

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

print(a\_inv)

**7. Визначити ранг матриці: 1)**

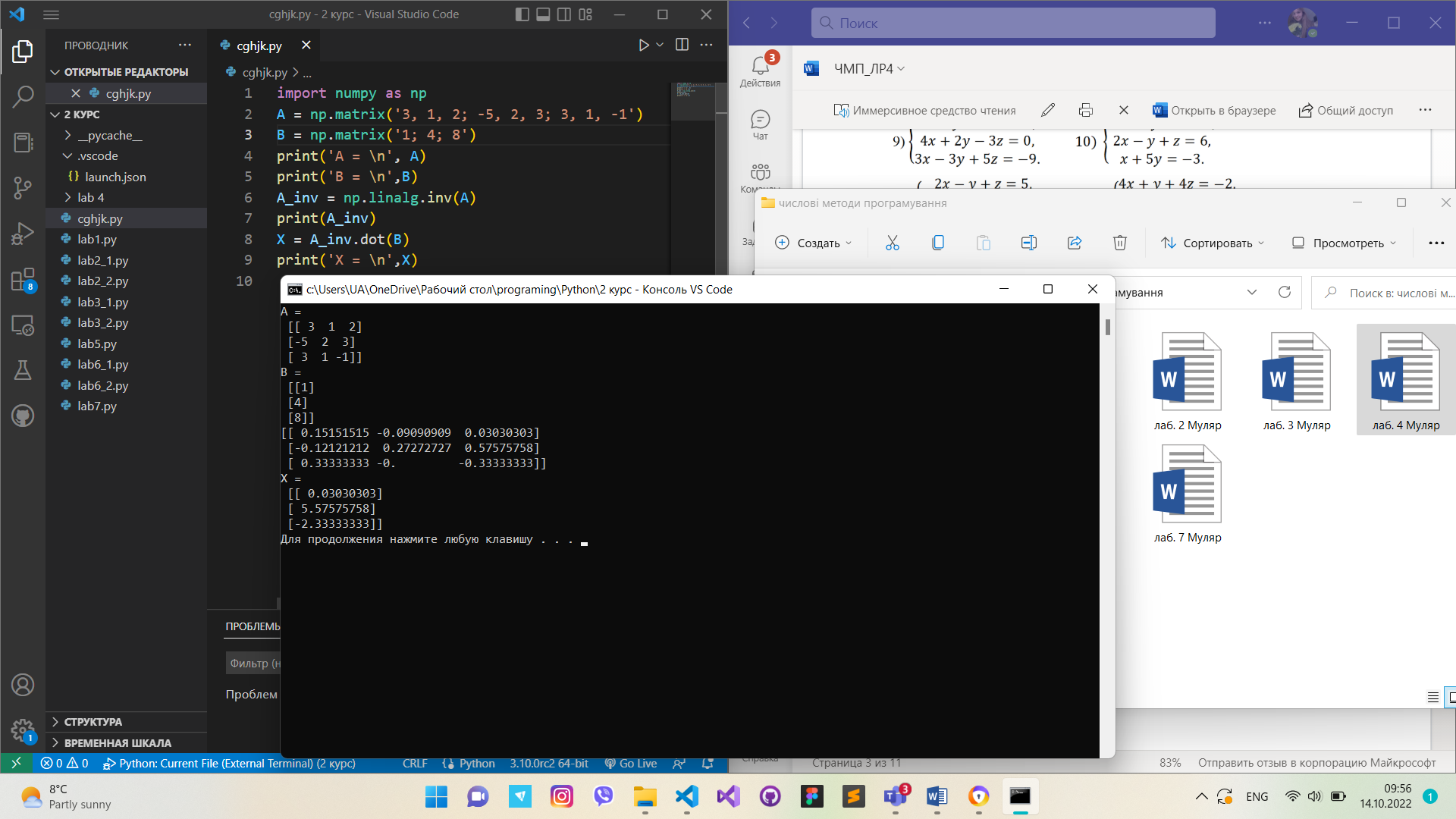
import numpy as np

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

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

print(rank)

**8. Розв’язати систему лінійних рівнянь матричним методом:**

import numpy as np

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

B = np.matrix('1; 4; 8')

print('A = \n', A)

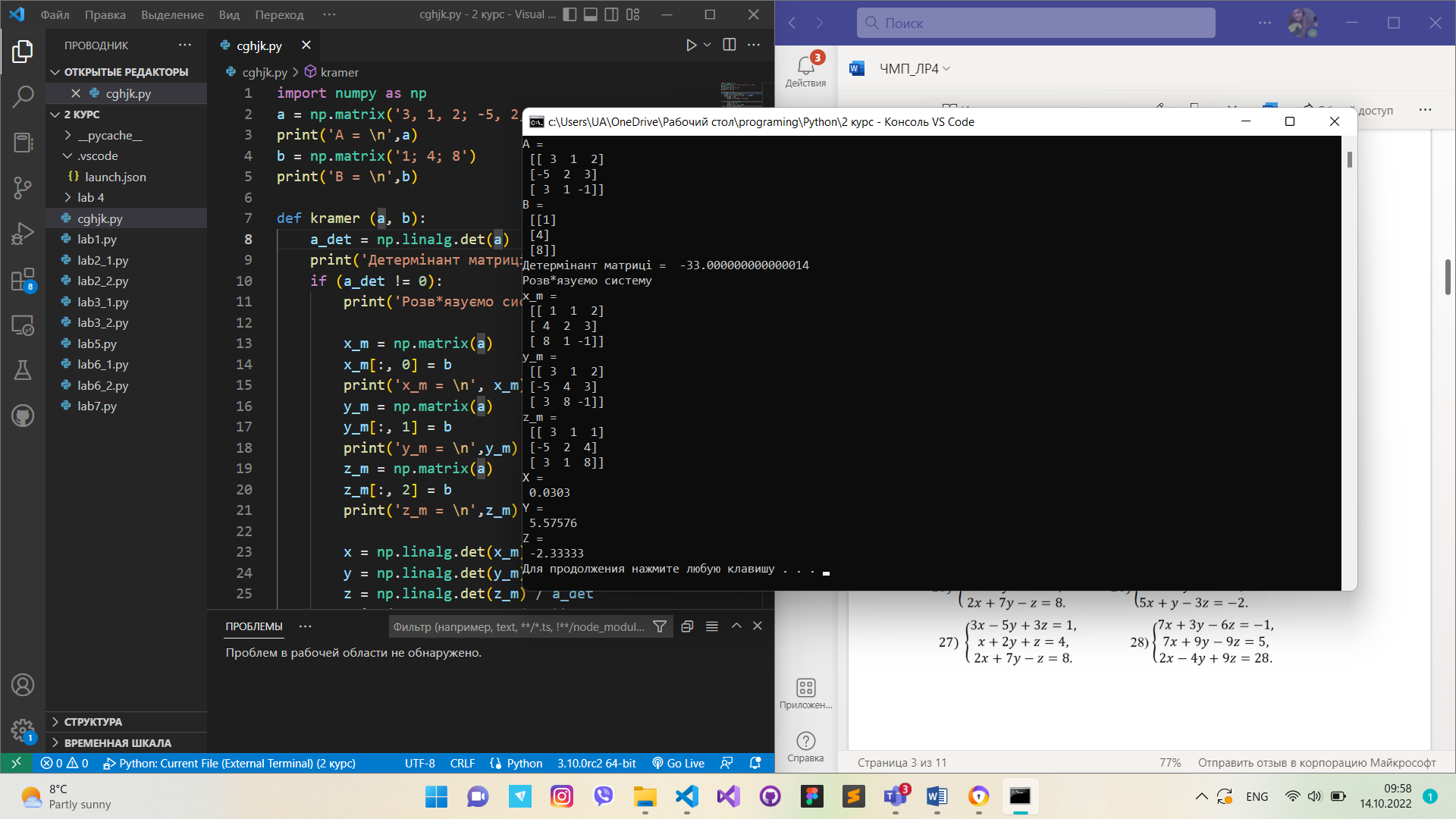
print('B = \n',B)

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

print(A\_inv)

X = A\_inv.dot(B)

print('X = \n',X)

**методом Крамера**

import numpy as np

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

print('A = \n',a)

b = np.matrix('1; 4; 8')

print('B = \n',b)

def kramer (a, b):

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

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

if (a\_det != 0):

print('Розв\*язуємо систему')

x\_m = np.matrix(a)

x\_m[:, 0] = b

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

y\_m = np.matrix(a)

y\_m[:, 1] = b

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

z\_m = np.matrix(a)

z\_m[:, 2] = b

print('z\_m = \n',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 = \n', round(x,5))

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

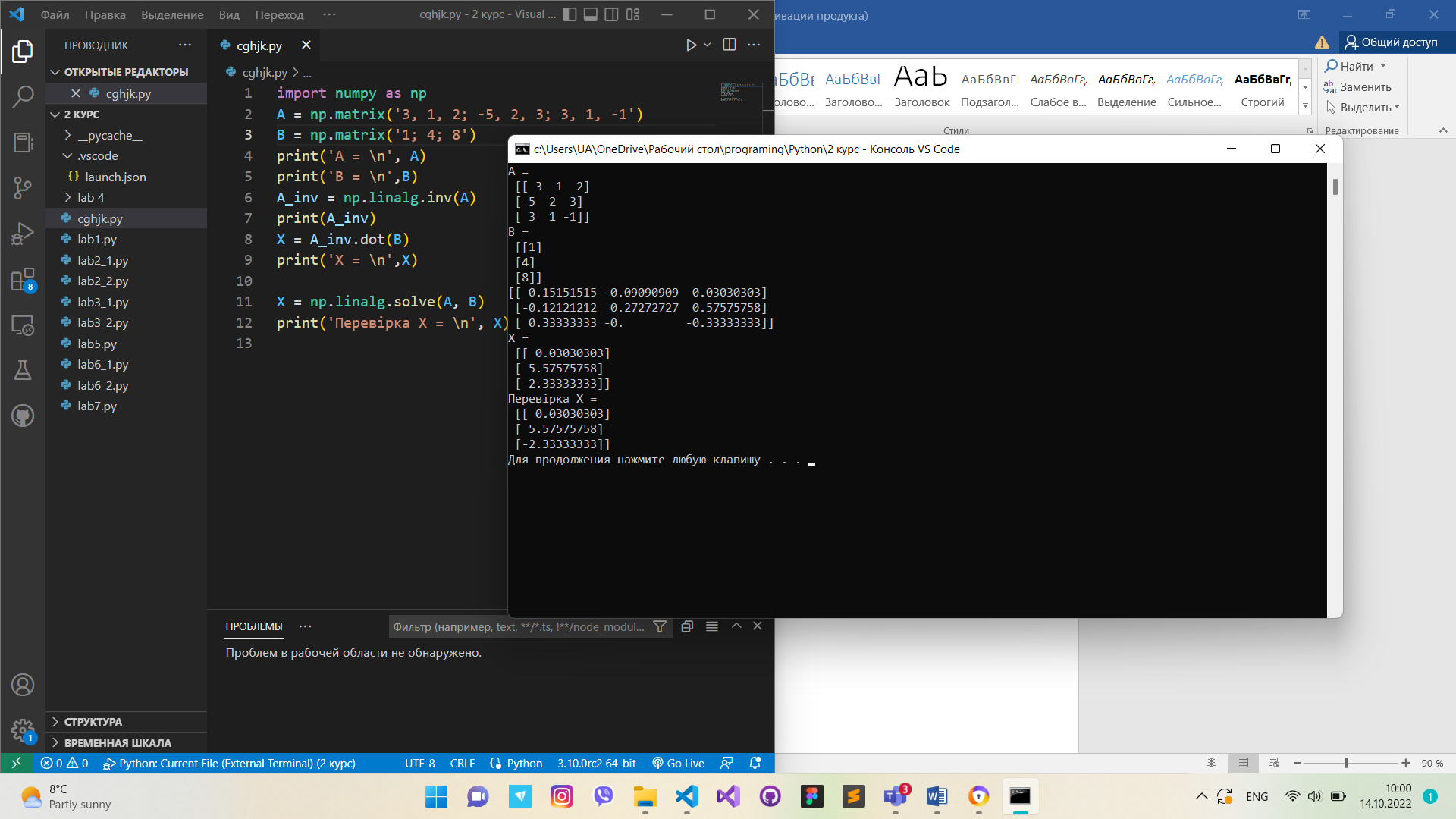
print('Z = \n', round(z,5))

else:

print('Розв\*язків немає')

kramer(a,b)

**9. Перевірка за допомогою методом solve() пакету linalg**

import numpy as np

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

B = np.matrix('1; 4; 8')

print('A = \n', A)

print('B = \n',B)

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

print(A\_inv)

X = A\_inv.dot(B)

print('X = \n',X)

X = np.linalg.solve(A, B)

print('Перевірка X = \n', X)

**10.**

**2. Створіть прямокутну матрицю A з N рядками та стовпцями M з випадкових елементів. Знайдіть найнижче значення серед середніх значень для кожного рядка матриці.**

import numpy as np

import pandas as pd

a = np.random.randint(-9, 9, (4, 6))

print("a = \n", a)

b = [\*map(sum, zip(\*a))]

print('Середнє арифметичне кожного стовпця = ', b)

min = min(b)

print('Мінімальне значення = ' , min)