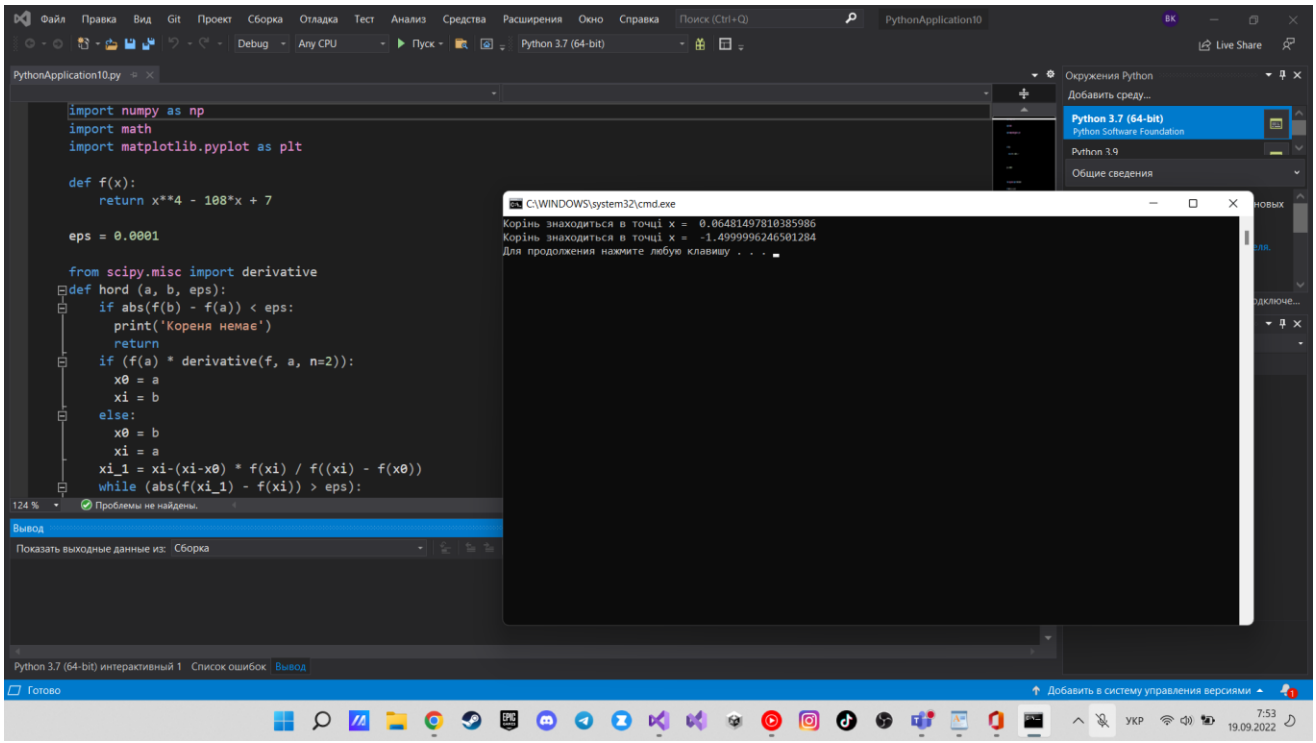
Завдання 1

1. Знайти матрицю $AB-BA$:

```
import numpy as np
import math
import matplotlib.pyplot as plt


def f(x):
    return x**4 - 108*x + 7


eps = 0.0001


from scipy.misc import derivative
def hord (a, b, eps):
    if abs(f(b) - f(a)) < eps:
      print('Кореня немає')
      return
    if (f(a) * derivative(f, a, n=2)):
      x0 = a
      xi = b
    else:
      x0 = b
      xi = a
    xi_1 = xi-(xi-x0) * f(xi) / f((xi) - f(x0))
    while (abs(f(xi_1) - f(xi)) > eps):
      xi = xi_1
      xi_1 = xi-(xi - x0) * f(xi) / (f(xi) - f(x0))


    else:
        print(f'Корінь знаходиться в точці x = ', xi_1)


hord(0.5, -2.5, 0.0001)
```

hord(3.5, -1.5, 0.0001)



Завдання 2

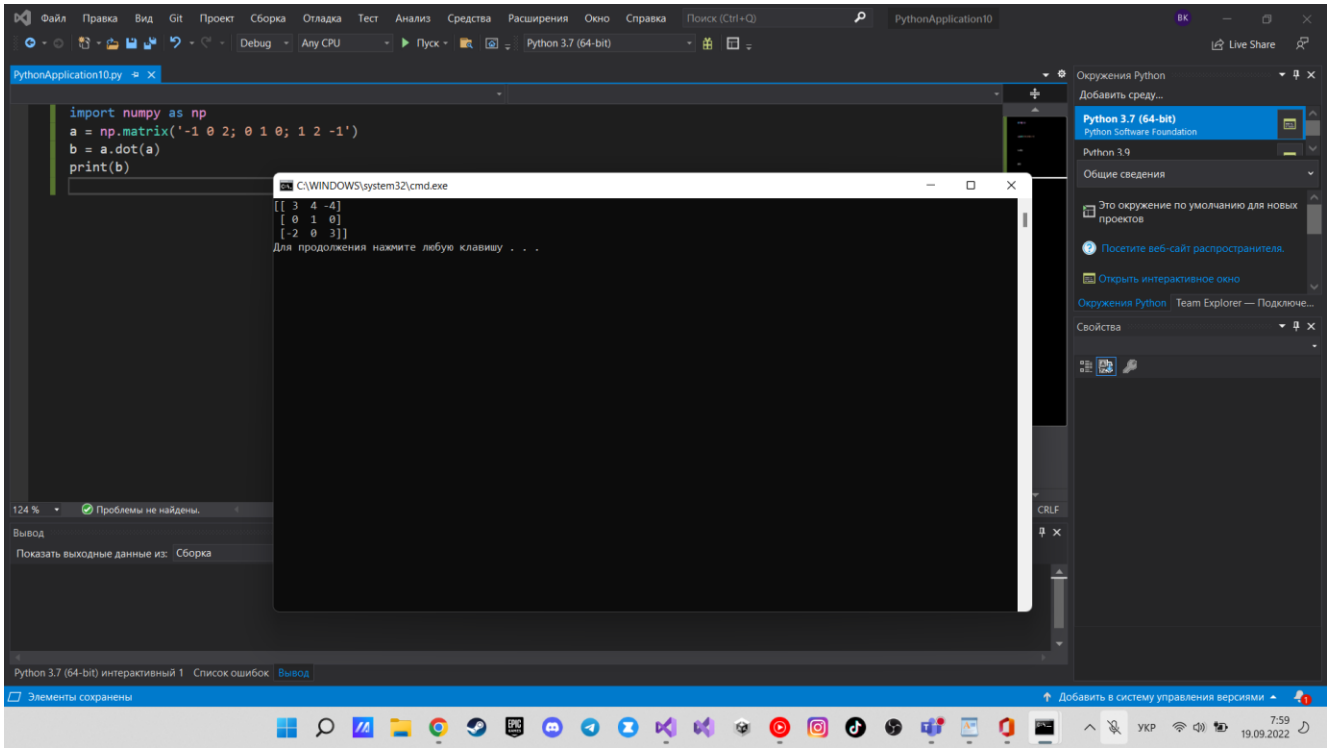
Піднести матриці до степеня:

import numpy as np

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

b = a.dot(a)

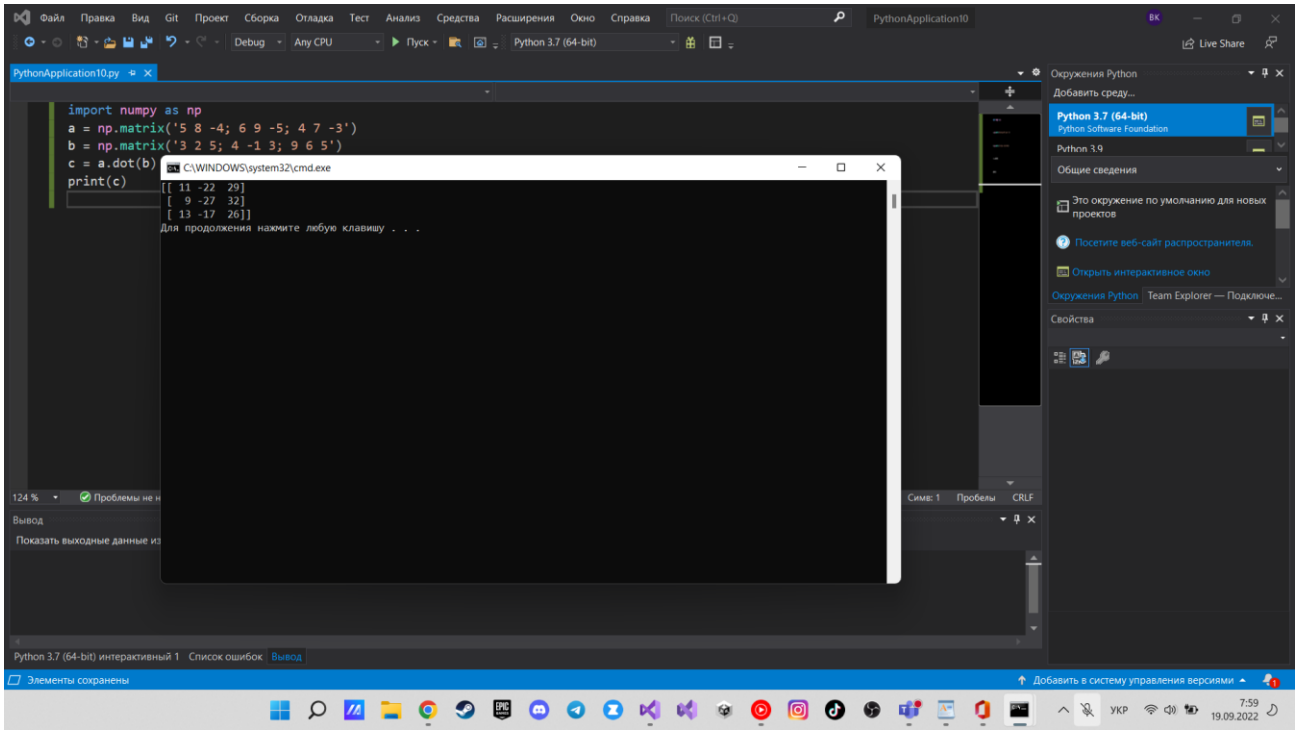
print(b)



Завдання 3

3. Знайти добуток матриць:

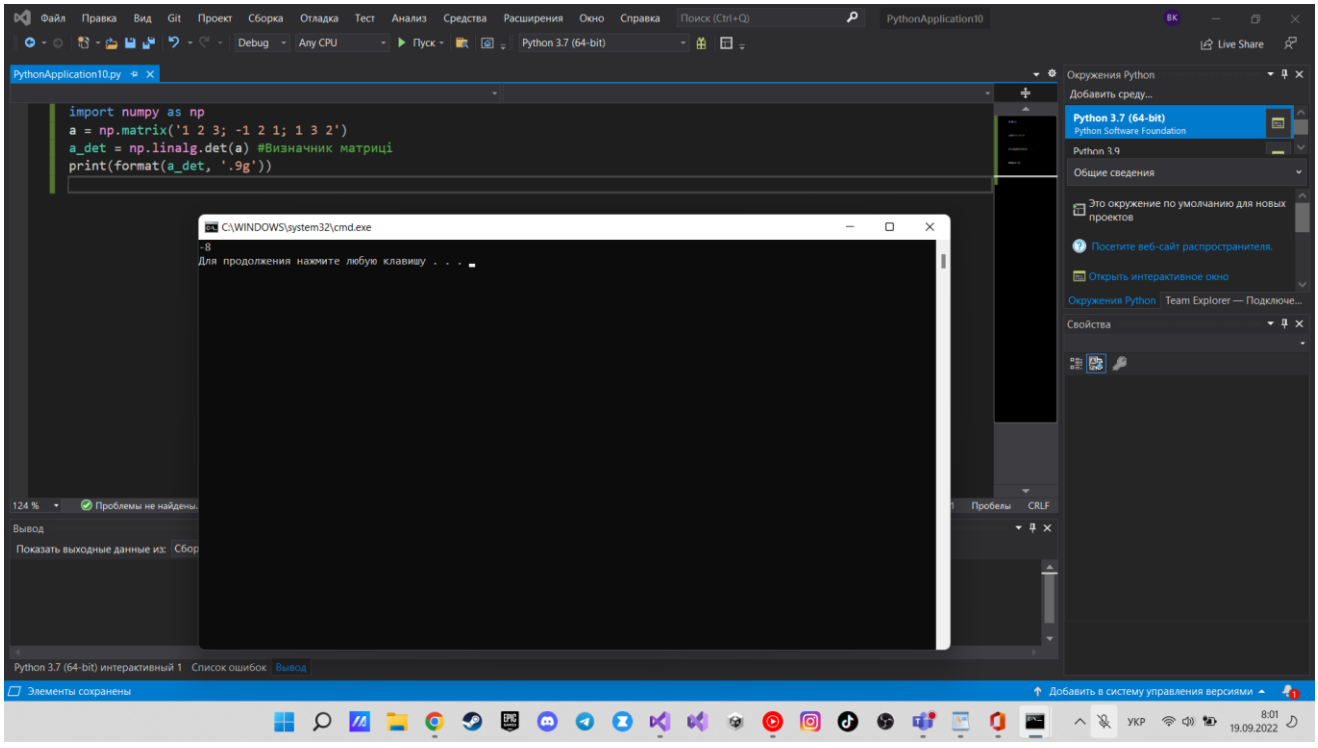
```
import numpy as np

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)
```



Заданння 4

4. Обчислити визначники:

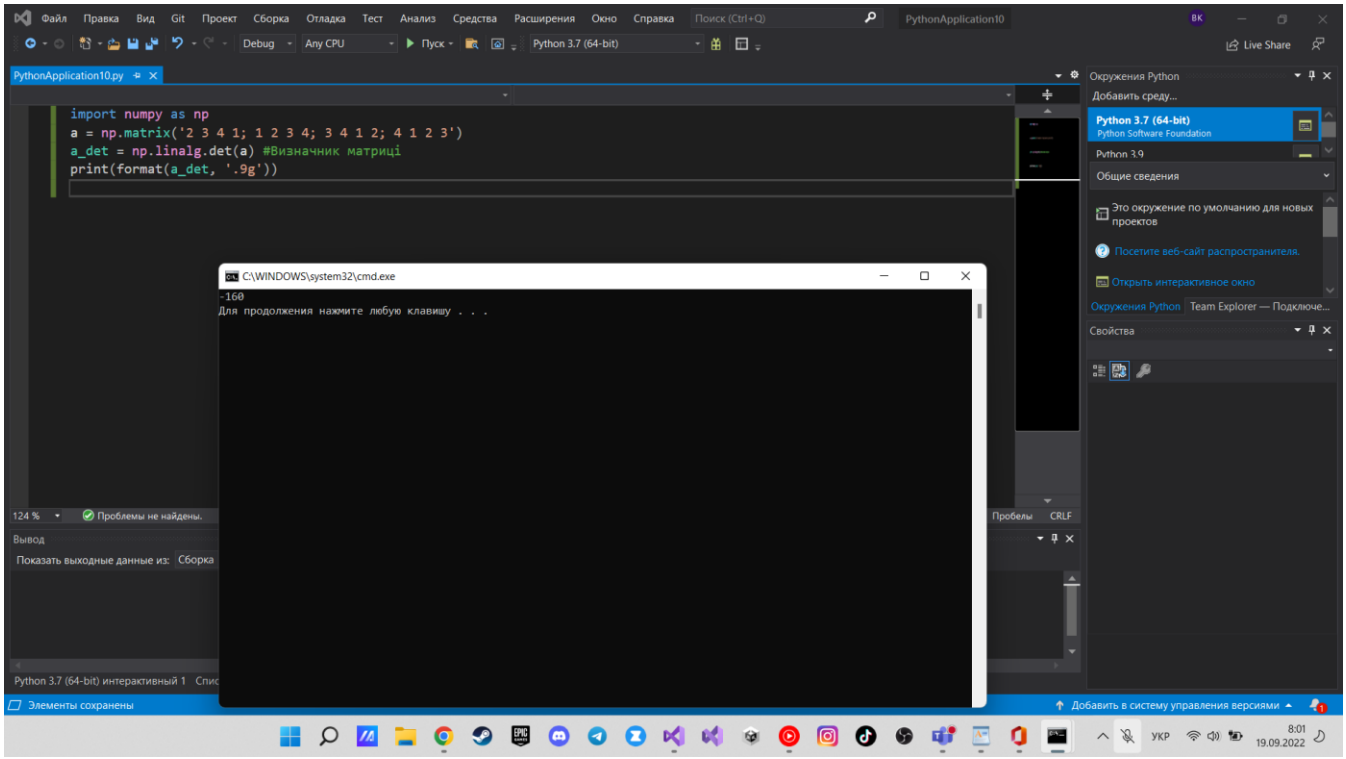
```
import numpy as np

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

a_det = np.linalg.det(a) #Визначник матриці

print(format(a_det, '.9g'))
```



Завдання 5

5. Обчислити визначники:

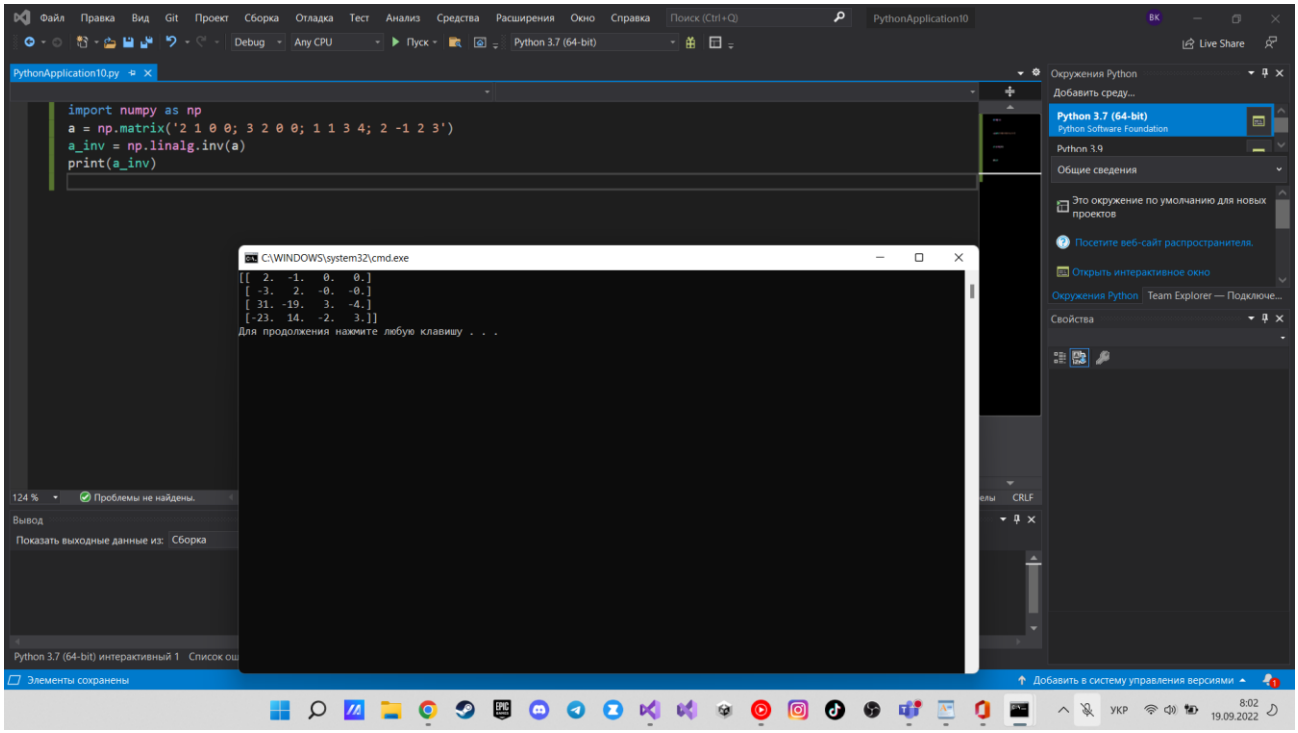
```
import numpy as np
a = np.matrix('2 3 4 1; 1 2 3 4; 3 4 1 2; 4 1 2 3')
a_det = np.linalg.det(a) #Визначник матриці
print(format(a_det, '.9g'))
```



Завдання 6

Знайти обернену матрицю до матриць:

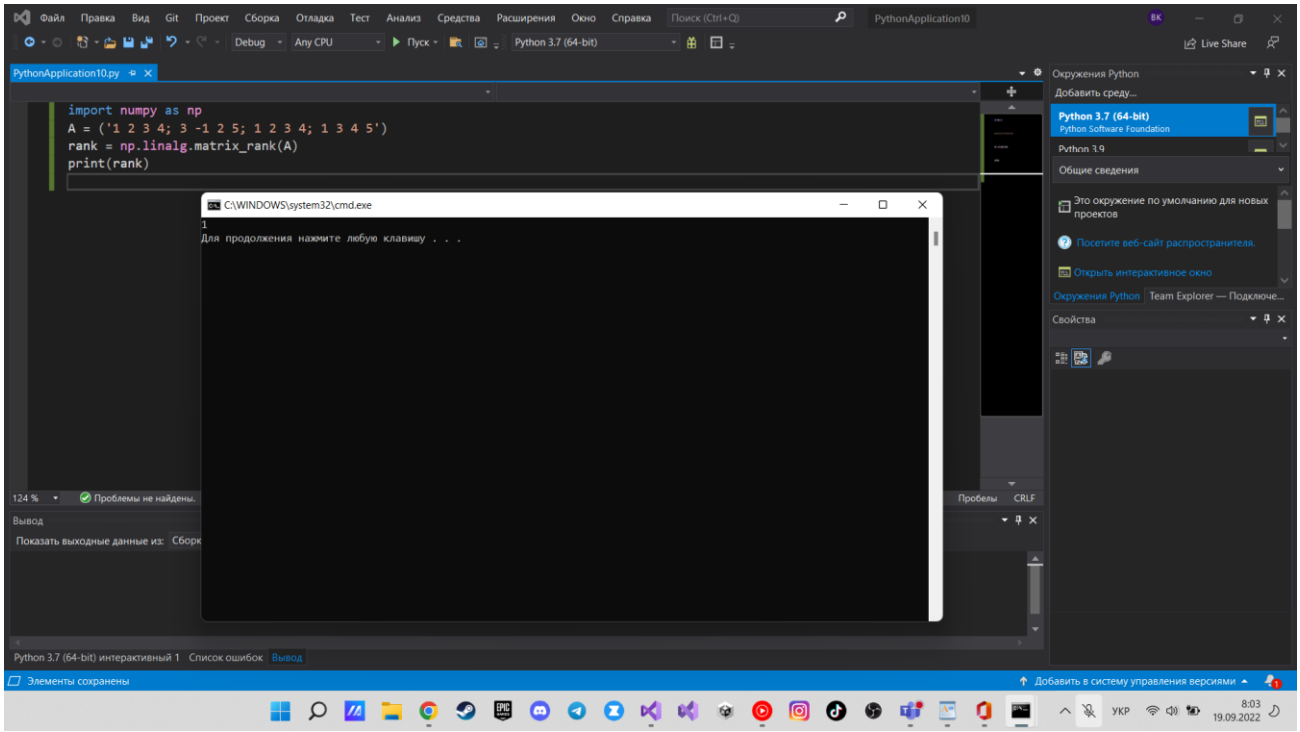
```
import numpy as np
a = np.matrix('2 1 0 0; 3 2 0 0; 1 1 3 4; 2 -1 2 3')
a_inv = np.linalg.inv(a)
print(a_inv)
```



Завдання 7

7. Визначити ранг матриці:

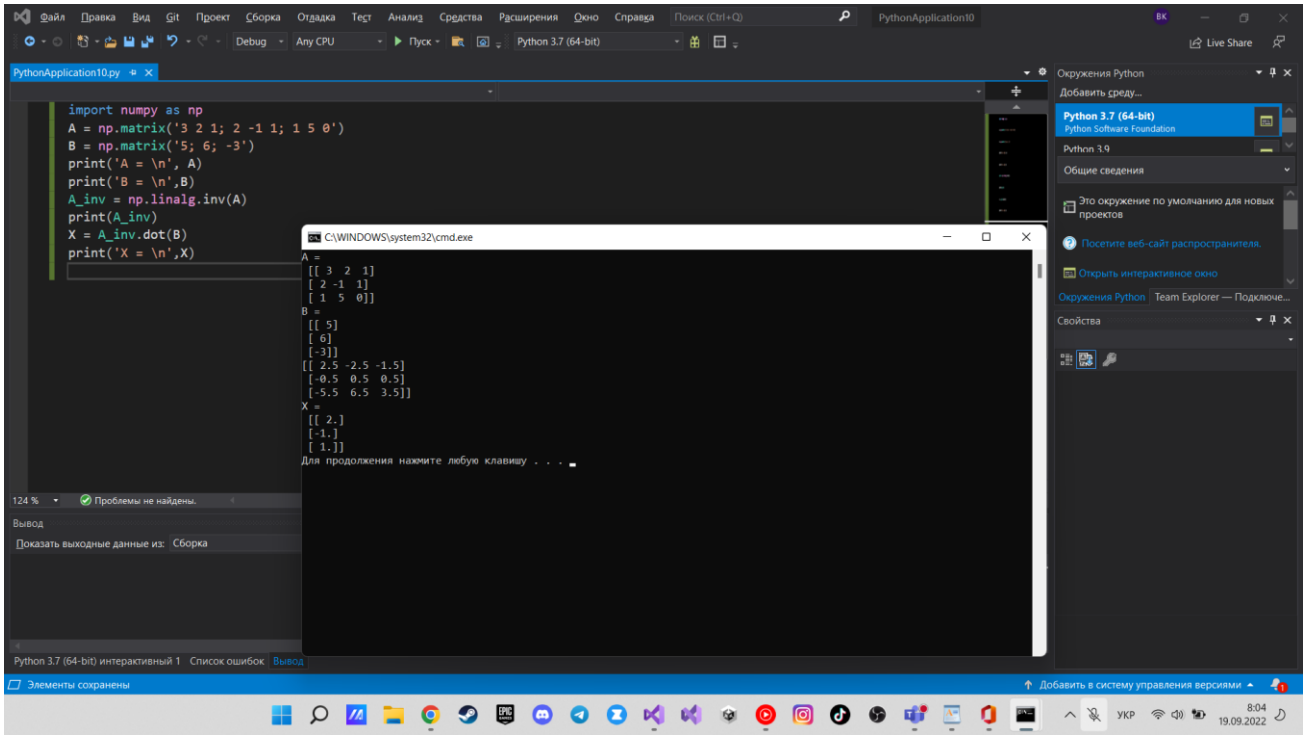
```
import numpy as np
A = ('1 2 3 4; 3 -1 2 5; 1 2 3 4; 1 3 4 5')
rank = np.linalg.matrix_rank(A)
print(rank)
```



Завдання 8

8. Розв’язати систему лінійних рівнянь матричним методом:

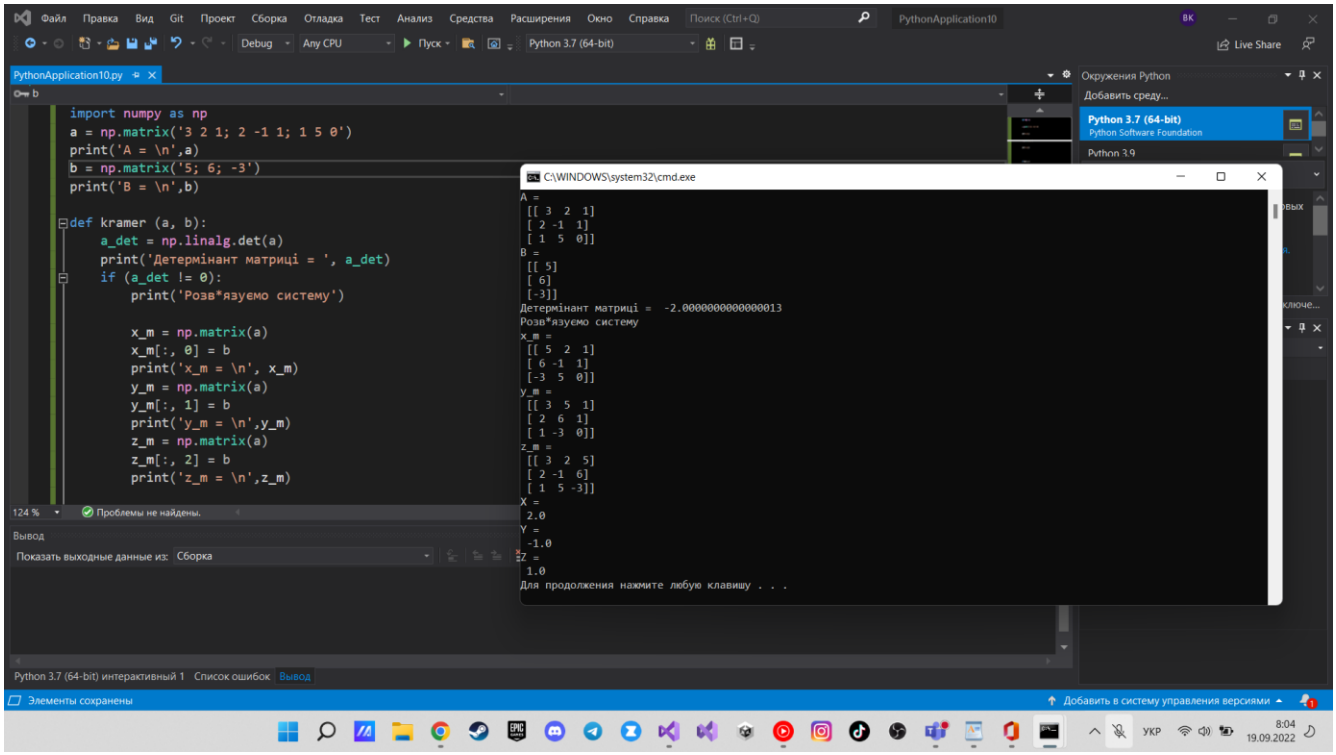
```
import numpy as np

A = np.matrix('3 2 1; 2 -1 1; 1 5 0')

B = np.matrix('5; 6; -3')

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 2 1; 2 -1 1; 1 5 0')

print('A = \n',a)

b = np.matrix('5; 6; -3')

print('B = \n',b)
```

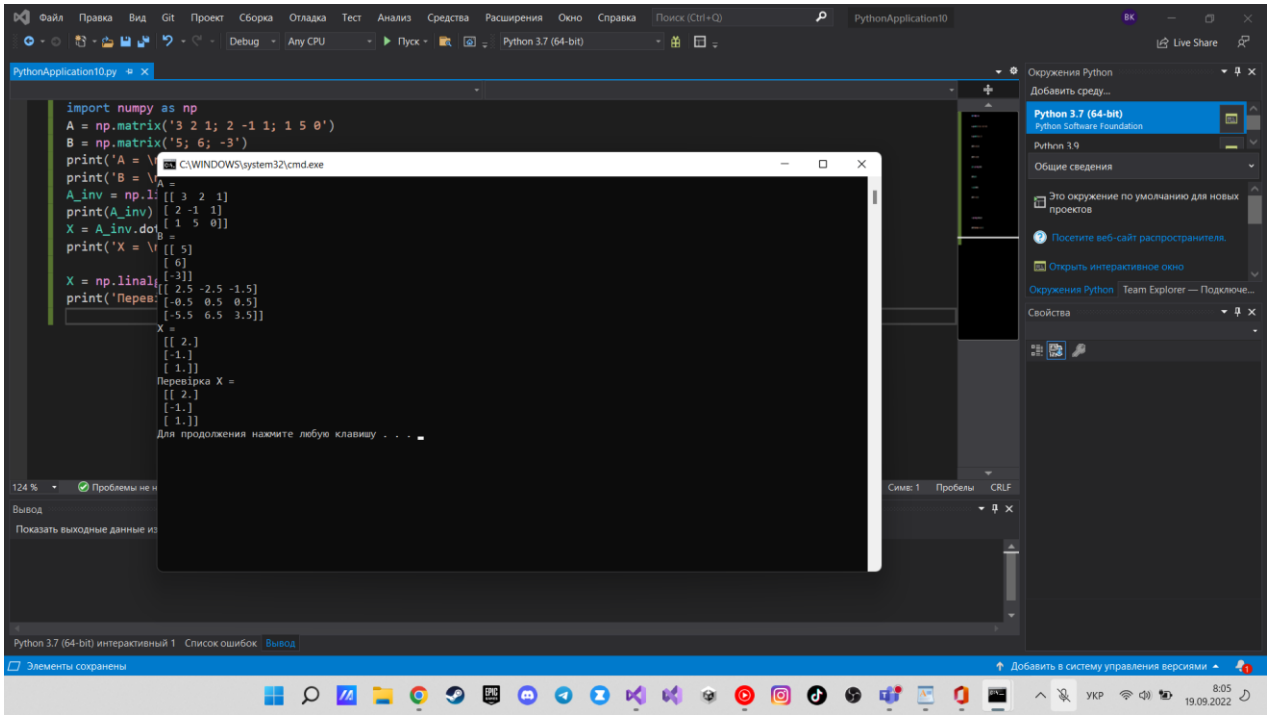
```
def kramer (a, b):
    a_det = np.linalg.det(a)
    print('Детермінант матриці = ', a_det)
    if (a_det != 0):
        print('Розв*язуємо систему')


        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)
```



Завдання 9

9. Перевірка за допомогою методом solve() пакету linalg

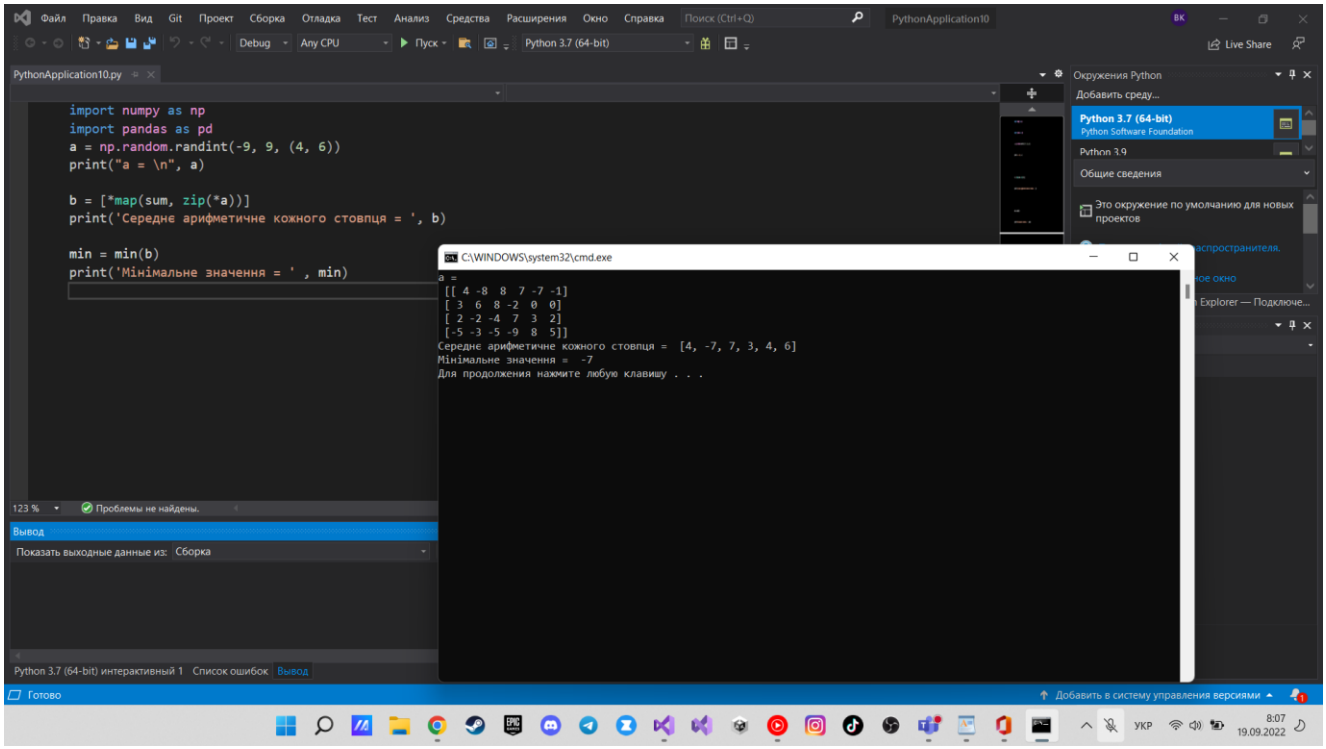
```
import numpy as np

A = np.matrix('3 2 1; 2 -1 1; 1 5 0')

B = np.matrix('5; 6; -3')

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)
```



Завдання 10

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)
```


Файл Правка Вид Git Проект Сборка Отладка Тест Анализ Средства Расширения Окно Справка
Debug
Any CPU
Пуск
Python 3.7 (64-bit)
Live Share
PythonApplication10.py
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)
Окружения Python
Добавить среду...
Python 3.7 (64-bit)
Python Software Foundation
Python 3.9
Общие сведения
Это окружение по умолчанию для новых проектов
C:\WINDOWS\system32\cmd.exe
a =
[[4 -8 8 7 -7 -1]
[3 6 8 -2 0 0]
[2 -2 -4 7 3 2]
[-5 -3 -5 -9 8 5]]
Середнє арифметичне кожного стовпця = [4, -7, 7, 3, 4, 6]
Мінімальне значення = -7
Для продолжения нажмите любую клавишу . . .
123 %
Проблемы не найдены.
Вывод
Показать выходные данные из: Сборка
Python 3.7 (64-bit) интерактивный 1
Список ошибок
Вывод
Готово
Добавить в систему управления версиями
УКР
8:07
19.09.2022