**Бігун Максим Сергійович, ФІТ 2-7, Варіант 2**

**Завдання:** Елементи лінійної алгебри. Розв’язання систем лінійних алгебраїчних рівнянь.

*Розв’язання:*

* 1. Знайти матрицю :

*Код:*

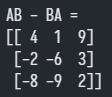
# 1.1.2

mat\_a = np.matrix([[2, 3, 1], [-1, 1, 0], [1, 2, -1]])

mat\_b = np.matrix([[1, 2, 1], [0, 1, 2], [3, 1, 1]])

print('AB - BA =')

print(mat\_a \* mat\_b - mat\_b \* mat\_a)



1.2. Піднесення матриці до степені:

*Код:*

# 1.2.2

mat\_a = np.matrix([[-1, 0, 2], [0, 1, 0], [1, 2, -1]])

print('A^2 =')

print(mat\_a \* mat\_a)

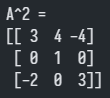
        else:

            mat\_c[i].append(mat\_a[i][j]\*mat\_a[j][i])

print('A^2:')

for row in mat\_c:

    print(row)



1.3. Знайти добуток матриць:

*Код:*

# 1.3.2

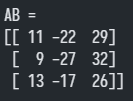
mat\_a = np.matrix([[5, 8, -4], [6, 9, -5], [4, 7, -3]])

mat\_b = np.matrix([[3, 2, 5], [4, -1, 3], [9, 6, 5]])

mat\_c = mat\_a \* mat\_b

print('AB =')

print(mat\_c)



1.4. Обчислити визначник:

*Код:*

# 1.4.2

mat\_a = np.matrix([[0, 2, 0], [3, 4, 5], [6, 7, 8]])

mat\_a\_det = np.linalg.det(mat\_a)

print(f'det(A) = {mat\_a\_det:.2f}')



1.5. Обчислити визначник:

*Код:*

# 1.5.2

mat\_a = np.matrix([[2, 3, 4, 1], [1, 2, 3, 4], [3, 4, 1, 2], [4, 1, 2, 3]])

mat\_a\_det = np.linalg.det(mat\_a)

print(f'det(A) = {mat\_a\_det:.2f}')



1.6. Знайти обернену матрицю до матриці:

*Код:*

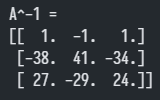
# 1.6.2

mat\_a = np.matrix([[2, 5, 7], [6, 3, 4], [5, -2, -3]])

mat\_a\_inv = np.linalg.inv(mat\_a)

print('A^-1 =')

print(mat\_a\_inv)



1.7. Визначити ранг матриці:

*Код:*

# 1.7.2

mat\_a = np.matrix([[1, -1, 3, 4], [0, -1, 2, 1], [1, 1, -1, 2], [2, 3, -5, 3]])

mat\_a\_rank = np.linalg.matrix\_rank(mat\_a)

print(f'rank(A) = {mat\_a\_rank}')



1.8. Розв’язати систему лінійних рівнянь методом Крамера, матричним методом:

*Код:*

# 1.8.2

def kramer(A, B):

    a\_det = np.linalg.det(A)

    if a\_det != 0:

        x\_mat = np.matrix(A)

        x\_mat[:, 0] = B

        y\_mat = np.matrix(A)

        y\_mat[:, 1] = B

        z\_mat = np.matrix(A)

        z\_mat[:, 2] = B

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

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

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

        return x, y, z

    else:

        return None

mat\_a = np.matrix([[2, -1, 1], [3, 2, 2], [1, -2, 1]])

mat\_b = np.matrix([[2], [-2], [1]])

x, y, z = kramer(mat\_a, mat\_b)

print(f'x= {x: .2f}\ny= {y: .2f}\nz= {z: .2f}')



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

*Код:*

# 2.6

import random

n = int(input('Enter n: '))

m = int(input('Enter m: '))

mat\_a = np.matrix([[random.randint(-10, 10) for j in range(m)] for i in range(n)])

print('A =')

print(mat\_a)

avg\_rows = np.mean(mat\_a, axis=1)

min\_row = np.argmin(avg\_rows)

print(f'Row with the lowest average: {min\_row+1}')

print(f'Average: {avg\_rows[min\_row, 0]:.2f}')

