Міністерство освіти та науки України

Національний технічний університет України

«Київський Політехнічний Інститут ім. Ігоря Сікорського»

ФТІ

Кафедра ФТЗЗІ

**Лабораторна робота №7**

з дисципліни: «Програмування 4»

на тему:

**«Обробка символьної інформації»**

**Варіант 21**

Виконав:

Ст. гр. ФЕ-81

Пелих Валентин

Перевірив:

доцент Прогонов Д.О.

Київ 2020

**Мета роботи:** Оволодіти методами та функціями обробки символьної інформації.

**Порядок виконання роботи:**  
1. Проаналізувати умову задачі.

2. Розробити алгоритм та створити програму розв’язання задачі згідно з номером варіанту.

3. Результати роботи оформити протоколом.

**Варіанти завдань:**

21. Задано дві матриці цілих чисел розмірністю  та  відповідно. Виконати перемноження цих матриць, якщо це не можливо, зменшити їх таким чином, щоб це можна було зробити. Обчислити кількість елементів отриманої після перемноження матриці. Представити її у вигляді стрингів.

**Код реалізації:**

import numpy as np

import random

print("Enter Two matrix.", "\n")

print("Firts matrix: m x n")

print("Second matrix: k x l")

# Only non-negative number

m = abs(int(input(" Enter m: ")))

n = abs(int(input(" Enter n: ")))

k = abs(int(input(" Enter k: ")))

l = abs(int(input(" Enter l: ")))

# Use generator to fill random numbers

M1 = [[ random.randint(-10, 10) for i in range(0, n)] for j in range(0, m)]

M2 = [[ random.randint(-10, 10) for i in range(0, l)] for j in range(0, k)]

M1 = np.array(M1)

M2 = np.array(M2)

print(M1, "\n")

print(M2, "\n")

def Mul(M1, M2, c):

if (c == 1):

print("Matrix multiplication by number!")

print("First matrix M1 is aproximated to number.")

M3 = np.multiply(M1, M2)

print("Multiplication is done!")

return M3

elif (c == 2):

print("Matrix multiplication by number!")

print("Second matrix M2 is aproximated to number.")

M3 = np.multiply(M1, M2)

print("Multiplication is done!")

return M3

else:

print("Calculating...")

M3 = np.dot(M1, M2)

print("Matrix-Matrix multiplication is done!")

return M3

if (n == k):

print("N is equal to K.", "\n")

MulM = Mul(M1, M2, 0)

print(MulM)

elif (n > k):

print("N above K: n > k.", "\n")

delta = n - k

M1 = M1[:, :n-delta]

if (len(M1) == 1):

n\_case = 1; n = M1[0][0]

MulM = Mul(n, M2, n\_case)

print(MulM)

else:

MulM = Mul(M1, M2, 0)

print(MulM)

elif (n < k):

print("N less K: n < k.", "\n")

delta = k - n

M2 = M2[:k-delta, :]

if (len(M2) == 1):

n\_case = 2; n = M2[0][0]

MulM = Mul(M1, n, n\_case)

print(MulM)

else:

MulM = Mul(M1, M2, 0)

print(MulM)

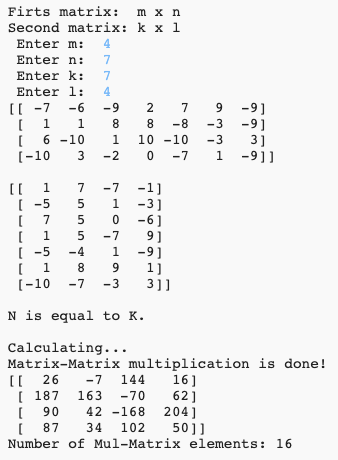
Size = MulM.size

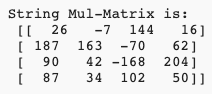
print("Number of Mul-Matrix elements:", Size, "\n")

MulMString = np.array2string(MulM)

print("String Mul-Matrix is:", "\n", MulMString)

**Приклади виконання програми:**

****

****