МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД

«УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ»

ФАКУЛЬТЕТ МАТЕМАТИКИ ТА ЦИФРОВИХ ТЕХНОЛОГІЙ СПЕЦІАЛЬНІСТЬ ПРИКЛАДНА МАТЕМАТИКА

«Звіт»

Лабораторна робота № 7

Виконав студент 1 курсу

Скорондяк Станіслав

Ужгород 2021

Завдання 1

# Визначити суму від’ємних елементів матриці на побічній діагоналі .  
  
a = [  
 [1, 4, 7, -8],  
 [1, 2, 4, 7],  
 [6, -24, 2, 1],  
 [-4, 2, 5, 1]  
]  
s = 0  
n = len(a)  
for i in range(n):  
 if a[i][n - (i + 1)] < 0:  
 s += a[i][n - (i + 1)]  
print(s)

Приклади:

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_1.py

-36

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_1.py

-130

Завдання 2

*"""  
Побудувати квадратну матрицю А, елементи якої задаються формулою:  
  
a[i][j]= i! , if (i+j)%2==0 парне  
 (i,j)=1,n  
 1\*\*2+2\*\*2+3\*\*2+...+j\*\*2 , (i+j)%2==1 непарне  
  
"""*# позначення  
"""  
n - (nxn), розмірність квадратної матриці;  
  
"""  
# введення розмірності  
  
n = int(input("n="))  
i = j = n  
  
def sum\_j(j):  
 return sum(map(lambda el: el \*\* 2, range(1, j + 1)))  
  
  
# Введення матриці  
def matrix\_a(arr):  
 for row in arr:  
 for el in row:  
 print(el, end=' ')  
 print()  
  
  
# Обчислення факторіалу  
def factorial(m):  
 if m == 0:  
 return 1  
 else:  
 return factorial(m - 1) \* m  
  
  
# введення елементів матриці  
  
def create\_a(num):  
 a = []  
 internal\_a = []  
 for j in range(num):  
 if (i + j) % 2 == 0:  
 internal\_a.append(factorial(i))  
 else:  
 internal\_a.append(sum\_j(j))  
 a.append(internal\_a)  
 return a  
  
  
# Створення матриці  
a = [[factorial(i) if (i + j) % 2 == 0 else sum\_j(j) for j in range(n)] for i in range(n)]  
  
  
z = matrix\_a(a)  
print(z)

Приклади:

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_2.py

n=6

1 1 1 14 1 55

0 1 5 1 30 1

2 1 2 14 2 55

0 6 5 6 30 6

24 1 24 14 24 55

0 120 5 120 30 120

None

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_2.py

n=2

1 1

0 1

None

Завдання 3

*"""  
Дано матрицю А .Перевірити, чи є дана матриця нижньою трикутною матрицею.  
  
"""*a = [  
 [5, 4, 7, -8],  
 [0, 9, 4, 7],  
 [0, 0, 8, 1],  
 [0, 6, 0, 0]  
]  
def is\_matrix\_triangle(a):  
 for i in range(len(a)):  
 for j in range(i):  
 if a[i][j] != 0:  
 return False  
 return True  
b = is\_matrix\_triangle(a)  
print(b)

Приклади:

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_3.py

False

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_3.py

True

Завдання 4

*"""  
13. Перемістити від’ємні елементи парних рядків наліво не міняючи їх порядок  
"""*import math  
a = [  
 [1, 4, 7, -8],  
 [1, 2, 4, 7],  
 [6, -24, 2, 1],  
 [-4, 2, 5, 1]  
]  
for i in range(0, len(a), 2):  
 for j in range(len(a[i])):  
 if a[i][j] < 0:  
 t = a[i][j-1]  
 a[i][j-1] = a[i][j]  
 a[i][j] = t  
print(a)

Приклади:

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_4.py

[[1, 4, -8, 7], [1, 2, 4, 7], [-24, 6, 2, 1], [-4, 2, 5, 1]]

Завдання 5

*"""  
Ущільнити задану матрицю, вилучаючи із неї рядки і стовпці, заповнені максимальним елементом.  
"""*a = [  
 [1, 4, 7, -8],  
 [1, 2, 4, 7],  
 [6, -24, 2, 1],  
 [-4, 2, 5, 1]  
]  
  
max\_num = a[0][0]  
for i in range(len(a)):  
 for j in range(len(a[i])):  
 if max\_num < a[i][j]:  
 max\_num = a[i][j]  
 k=j  
for el in a:  
 el.pop(k)  
print(a)

Приклади :

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_5.py

[[1, 4, -8], [1, 2, 7], [6, -24, 1], [-4, 2, 1]]

Завдання 6

*"""  
Дана цілочислова квадратна матриця. Визначити кількість рядків, у яких елементи упорядковані за зростанням  
"""*a = [  
 [1, 4, 7, -8],  
 [1, 2, 4, 7],  
 [6, -24, 2, 1],  
 [-4, 2, 5, 1]  
 #[True, True, False]  
]  
  
s = 0  
for i in range(len(a)):  
 ordered\_iter\_obj = map(lambda j: a[i][j - 1] <= a[i][j], range(1, len(a[i])))  
 is\_ordered = all(ordered\_iter\_obj)  
  
 # for j in range(1, len(a[i])):  
 # if a[i][j-1] > a[i][j]:  
 # is\_ordered = False  
 # break  
 if is\_ordered:  
 s += 1  
print(s)

Приклад:

C:\Users\User\AppData\Local\Programs\Python\Python310\python.exe C:/Users/User/Desktop/Python/Labka\_7/Labka\_7\_6.py

1