МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота №3

з дисципліни «Об'єктно-орієнтоване програмування СУ»

Тема:"Структурування програм з використанням функцій"

Виконав студент гр. 321

Новіков Олексій

(підпис, дата) (П.І.Б.)

Перевірив

к.т.н., доц. О. В. Гавриленко

ас.  В. О. Білозерський

2023

МЕТА РОБОТИ

Вивчити теоретичний матеріал із синтаксису визначення і виклику функцій та особливостей послідовностей у Python, а також документацію бібліотеки numpy; отримати навички реалізації бібліотеки функцій з параметрами, що структурують вирішення завдань «згори – до низу».

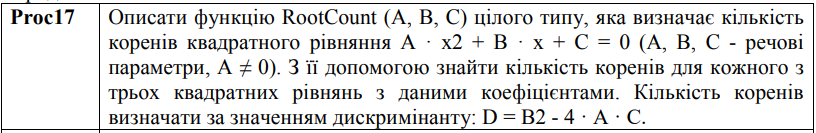
ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Описати функцію відповідно до варіанту. Для виклику функції (друга частина задачі) описати іншу функцію, що на вході має список вхідних даних і повертає список вихідних даних. Введення даних, виклик функції та виведення результатів реалізувати в третій функції без параметрів. Завдання наведено в табл.1.

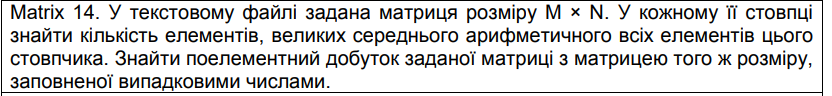
Завдання 2. Розробити дві вкладені функції для вирішення задачі обробки двовимірних масивів відповідно до варіанту: зовнішня – без параметрів, внутрішня має на вході ім’я файлу з даними, на виході – підраховані параметри матриці (перша частина задачі) та перетворену матрицю (друга частина задачі). Завдання представлено в табл.2.

Хід роботи

Завдання 1



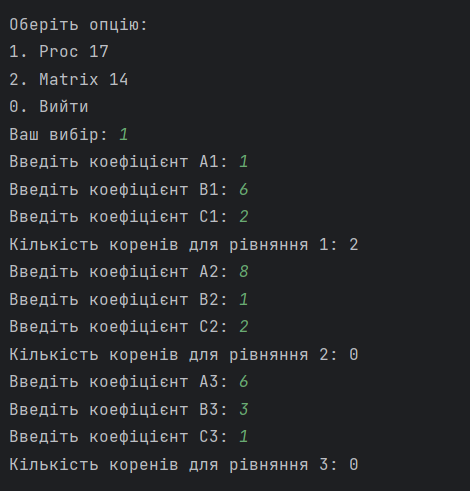
Завдання 2

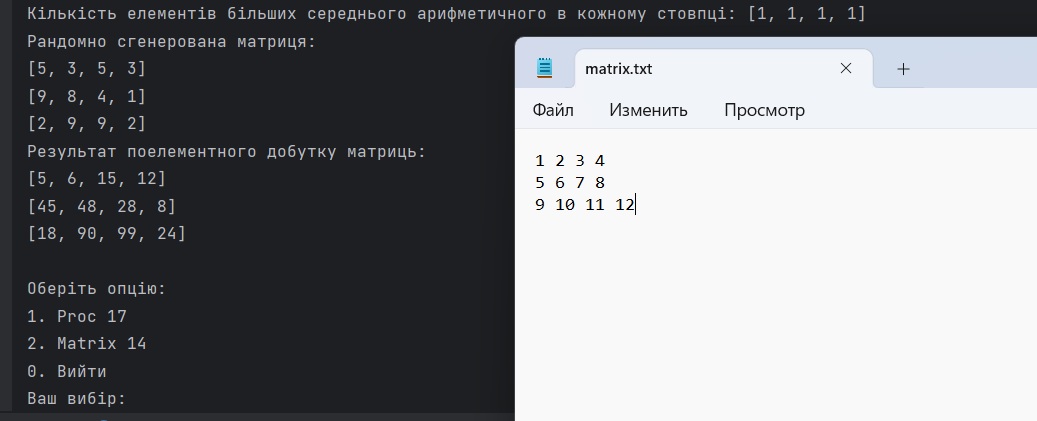


Лістинг коду

import random  
def RootCount(A, B, C):  
 # Визначаємо дискримінант  
 D = B \*\* 2 - 4 \* A \* C  
  
 # Перевіряємо значення дискримінанта  
 if D > 0:  
 # Два корені: один додатний, інший від'ємний  
 return 2  
 elif D == 0:  
 # Один подвійний корінь  
 return 1  
 else:  
 # Дискримінант від'ємний, немає дійсних коренів  
 return 0  
def proc17():  
 # Зчитування коефіцієнтів з консолі  
 A1 = float(input("Введіть коефіцієнт A1: "))  
 B1 = float(input("Введіть коефіцієнт B1: "))  
 C1 = float(input("Введіть коефіцієнт C1: "))  
  
 # Визначення кількості коренів і виведення результату  
 count = RootCount(A1, B1, C1)  
 print("Кількість коренів для рівняння 1:", count)  
 # Зчитування коефіцієнтів з консолі  
 A2 = float(input("Введіть коефіцієнт A2: "))  
 B2 = float(input("Введіть коефіцієнт B2: "))  
 C2 = float(input("Введіть коефіцієнт C2: "))  
  
 # Визначення кількості коренів і виведення результату  
 count = RootCount(A2, B2, C2)  
 print("Кількість коренів для рівняння 2:", count)  
 # Зчитування коефіцієнтів з консолі  
 A3 = float(input("Введіть коефіцієнт A3: "))  
 B3 = float(input("Введіть коефіцієнт B3: "))  
 C3 = float(input("Введіть коефіцієнт C3: "))  
  
 # Визначення кількості коренів і виведення результату  
 count = RootCount(A3, B3, C3)  
 print("Кількість коренів для рівняння 3:", count)  
def read\_matrix\_from\_file(filename):  
 matrix = []  
 with open(filename, 'r') as file:  
 for line in file:  
 row = [int(x) for x in line.strip().split()]  
 matrix.append(row)  
 return matrix  
  
# Функція для знаходження середнього арифметичного стовпця  
def average\_of\_column(matrix, col):  
 column = [row[col] for row in matrix]  
 return sum(column) / len(column)  
  
# Функція для знаходження кількості елементів більших середнього арифметичного в стовпці  
def count\_elements\_greater\_than\_average(matrix, col):  
 avg = average\_of\_column(matrix, col)  
 column = [row[col] for row in matrix]  
 return len([x for x in column if x > avg])  
  
# Функція для створення матриці з випадковими числами того ж розміру  
def generate\_random\_matrix(rows, cols):  
 return [[random.randint(1, 10) for \_ in range(cols)] for \_ in range(rows)]  
  
# Функція для обчислення поелементного добутку двох матриць  
def elementwise\_multiply(matrix1, matrix2):  
 result = []  
 for i in range(len(matrix1)):  
 row = [matrix1[i][j] \* matrix2[i][j] for j in range(len(matrix1[0]))]  
 result.append(row)  
 return result  
def matrix14():  
 # Зчитування матриці з файлу  
 matrix\_filename = 'matrix.txt'  
 matrix = read\_matrix\_from\_file(matrix\_filename)  
  
 # Знаходимо кількість елементів більших середнього арифметичного в кожному стовпці  
 num\_cols = len(matrix[0])  
 counts = [count\_elements\_greater\_than\_average(matrix, col) for col in range(num\_cols)]  
  
 print("Кількість елементів більших середнього арифметичного в кожному стовпці:", counts)  
  
 # Генерація нової матриці з випадковими числами  
 random\_matrix = generate\_random\_matrix(len(matrix), num\_cols)  
 print("Рандомно сгенерована матриця:")  
 for row in random\_matrix:  
 print(row)  
 # Обчислення поелементного добутку матриць  
 result\_matrix = elementwise\_multiply(matrix, random\_matrix)  
  
 # Виведення результату  
 print("Результат поелементного добутку матриць:")  
 for row in result\_matrix:  
 print(row)  
if \_\_name\_\_ == "\_\_main\_\_":  
 while True:  
 print("\nОберіть опцію:")  
 print("1. Proc 17")  
 print("2. Matrix 14")  
 print("0. Вийти")  
  
 choice = input("Ваш вибір: ")  
  
 if choice == "1":  
 proc17()  
 elif choice == "2":  
 matrix14()  
 elif choice == "0":  
 break  
 else:  
 print("Невірний вибір. Виберіть 1, 2, 3 або 0.")

Результат виконання





Висновок: під час виконання цієї лабораторної роботи я вивчив структурування програм з використанням функцій