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 проектування СУ»

Тема: «Структурування програм з використанням функцій»

ХАІ.301 . 3.320.3 ЛР

Виконав студент гр. 320

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Семеняга Ігор\_\_\_

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. О. В. Гавриленко

(підпис, дата) (П.І.Б.)

2024

# 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). Для обробки масивів використати функції бібліотеки numpy. Завдання представлено в табл.2.

# ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі 1, №11

Вхідні дані (ім’я, опис, тип, обмеження):

a = float(input("A: ")) float

b = float(input("B: ")) float

c = float(input("C: ")) float

d = float(input("D: ")) float

Вихідні дані (ім’я, опис, тип):

print("Мінімум: {min\_val}, Максимум: {max\_val}")

Алгоритм вирішення

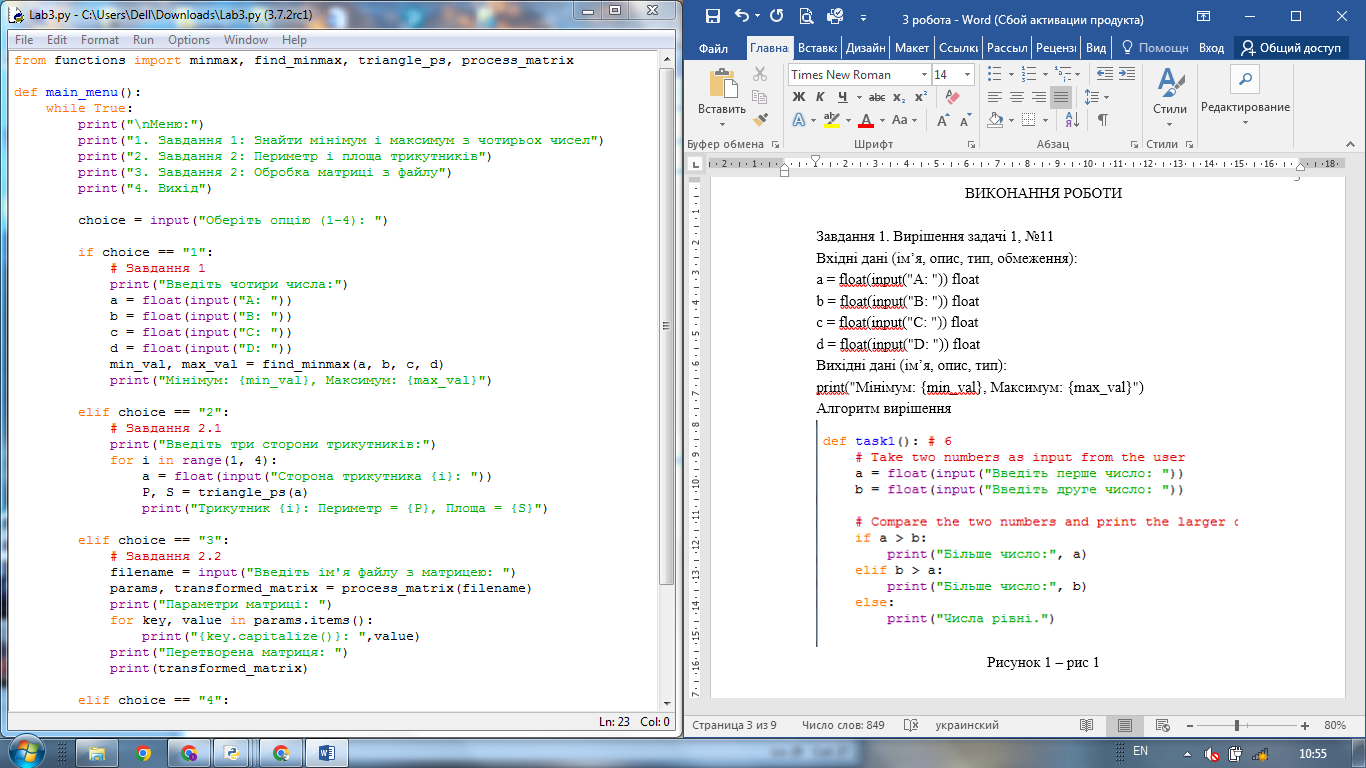


Рисунок 1 – рис 1

Лістинг коду вирішення задачі наведено в дод. А (стор. 6). Екран роботи програми показаний на рис. Б.7.

Завдання 2. Вирішення задачі 2, №4

Вхідні дані:

print("Введіть три сторони трикутників:")

for i in range(1, 4):

a = float(input(f"Сторона трикутника {i}: "))

P, S = triangle\_ps(a)

print(f"Трикутник {i}: Периметр = {P}, Площа = {S}") float

filename = input("Введіть ім'я файлу з матрицею: ") string

Вихідні дані:

print("Кількість точок у темно-зеленій області: ", count) int, string

Алгоритм вирішення

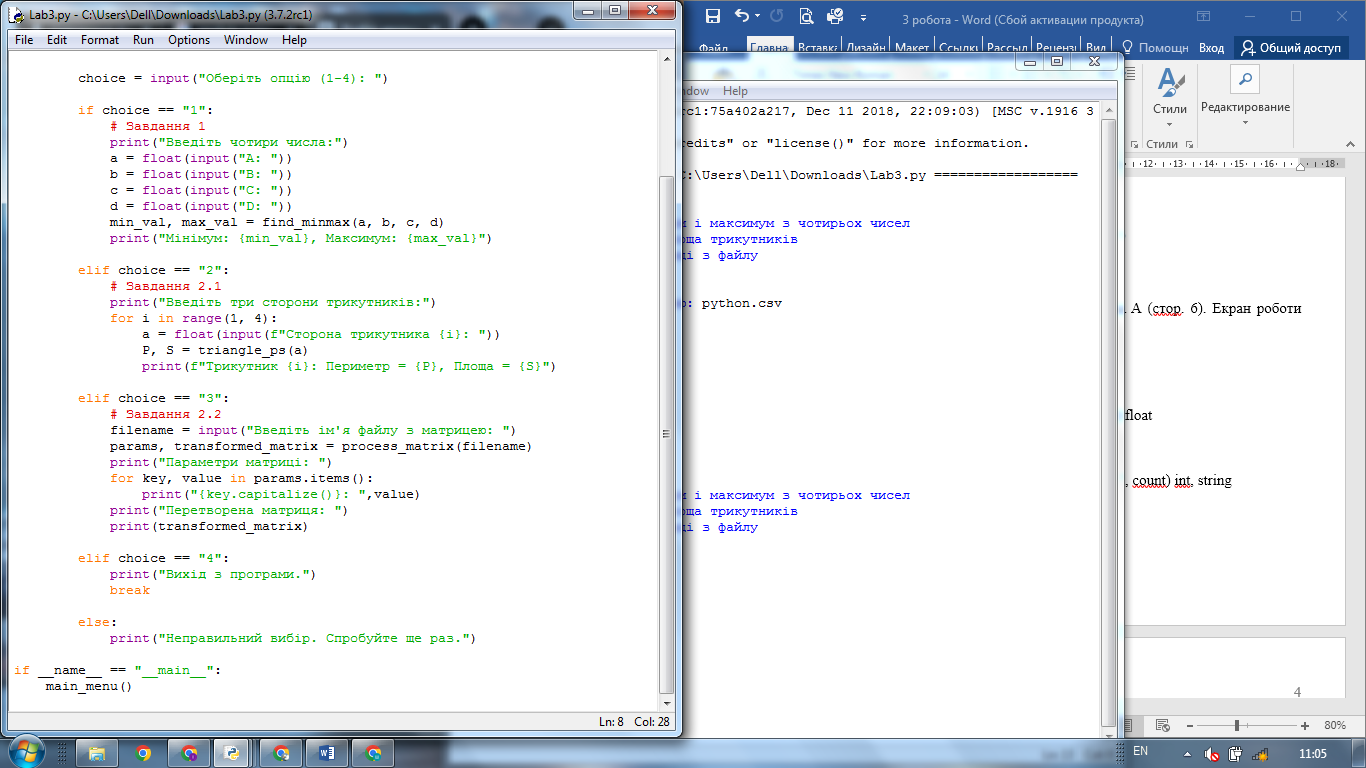


Рисунок 2 – рис 2

# ВИСНОВКИ

Було вивчено теоретичний матеріал із синтаксису визначення та виклику функцій у Python, а також особливості роботи з послідовностями. Ознайомлено з документацією бібліотеки NumPy для роботи з багатовимірними масивами та числовими обчисленнями. На практиці отримано навички створення бібліотеки функцій із параметрами, що забезпечують структурування вирішення завдань за підходом «згори – до низу».

ДОДАТОК А

Лістинг коду програми до задач 1-№11, 2-№4

from functions import minmax, find\_minmax, triangle\_ps, process\_matrix

def main\_menu():

while True:

print("\nМеню:")

print("1. Завдання 1: Знайти мінімум і максимум з чотирьох чисел")

print("2. Завдання 2: Периметр і площа трикутників")

print("3. Завдання 2: Обробка матриці з файлу")

print("4. Вихід")

choice = input("Оберіть опцію (1-4): ")

if choice == "1":

# Завдання 1

print("Введіть чотири числа:")

a = float(input("A: "))

b = float(input("B: "))

c = float(input("C: "))

d = float(input("D: "))

min\_val, max\_val = find\_minmax(a, b, c, d)

print("Мінімум: {min\_val}, Максимум: {max\_val}")

elif choice == "2":

# Завдання 2.1

print("Введіть три сторони трикутників:")

for i in range(1, 4):

a = float(input(f"Сторона трикутника {i}: "))

P, S = triangle\_ps(a)

print(f"Трикутник {i}: Периметр = {P}, Площа = {S}")

elif choice == "3":

# Завдання 2.2

filename = input("Введіть ім'я файлу з матрицею: ")

params, transformed\_matrix = process\_matrix(filename)

print("Параметри матриці: ")

for key, value in params.items():

print("{key.capitalize()}: ",value)

print("Перетворена матриця: ")

print(transformed\_matrix)

elif choice == "4":

print("Вихід з програми.")

break

else:

print("Неправильний вибір. Спробуйте ще раз.")

if \_\_name\_\_ == "\_\_main\_\_":

main\_menu()

import numpy as np

import math

# Завдання 1: Функція Minmax

def minmax(x, y):

"""

Функція знаходить мінімум і максимум з двох чисел.

Параметри:

x (float): перше число (вхід/вихід).

y (float): друге число (вхід/вихід).

"""

if x > y:

x, y = y, x

return x, y

# Завдання 1: Функція для знаходження мінімуму та максимуму з чотирьох чисел

def find\_minmax(a, b, c, d):

"""

Використовуючи чотири виклики minmax, знаходить мінімум і максимум з чотирьох чисел.

"""

a, b = minmax(a, b)

c, d = minmax(c, d)

a, c = minmax(a, c)

b, d = minmax(b, d)

return a, d # Мінімум і максимум

# Завдання 2: Функція TrianglePS

def triangle\_ps(a):

"""

Обчислює периметр і площу рівностороннього трикутника.

Параметри:

a (float): сторона трикутника.

Повертає:

P (float): периметр.

S (float): площа.

"""

P = 3 \* a

S = (a\*\*2 \* math.sqrt(3)) / 4

return P, S

# Завдання 2: Обробка матриць

def process\_matrix(filename):

"""

Зчитує матрицю з файлу, обчислює її параметри та перетворює.

Параметри:

filename (str): ім'я файлу з даними.

Повертає:

params (dict): параметри матриці.

transformed\_matrix (ndarray): перетворена матриця.

"""

# Зчитування матриці з файлу

matrix = np.loadtxt(filename, delimiter=",")

# Обчислення параметрів

params = {

"sum": np.sum(matrix),

"mean": np.mean(matrix),

"min": np.min(matrix),

"max": np.max(matrix)

}

# Перетворення матриці (транспонування)

transformed\_matrix = matrix.T

return params, transformed\_matrix

ДОДАТОК Б

Скрін-шоти вікна виконання програми

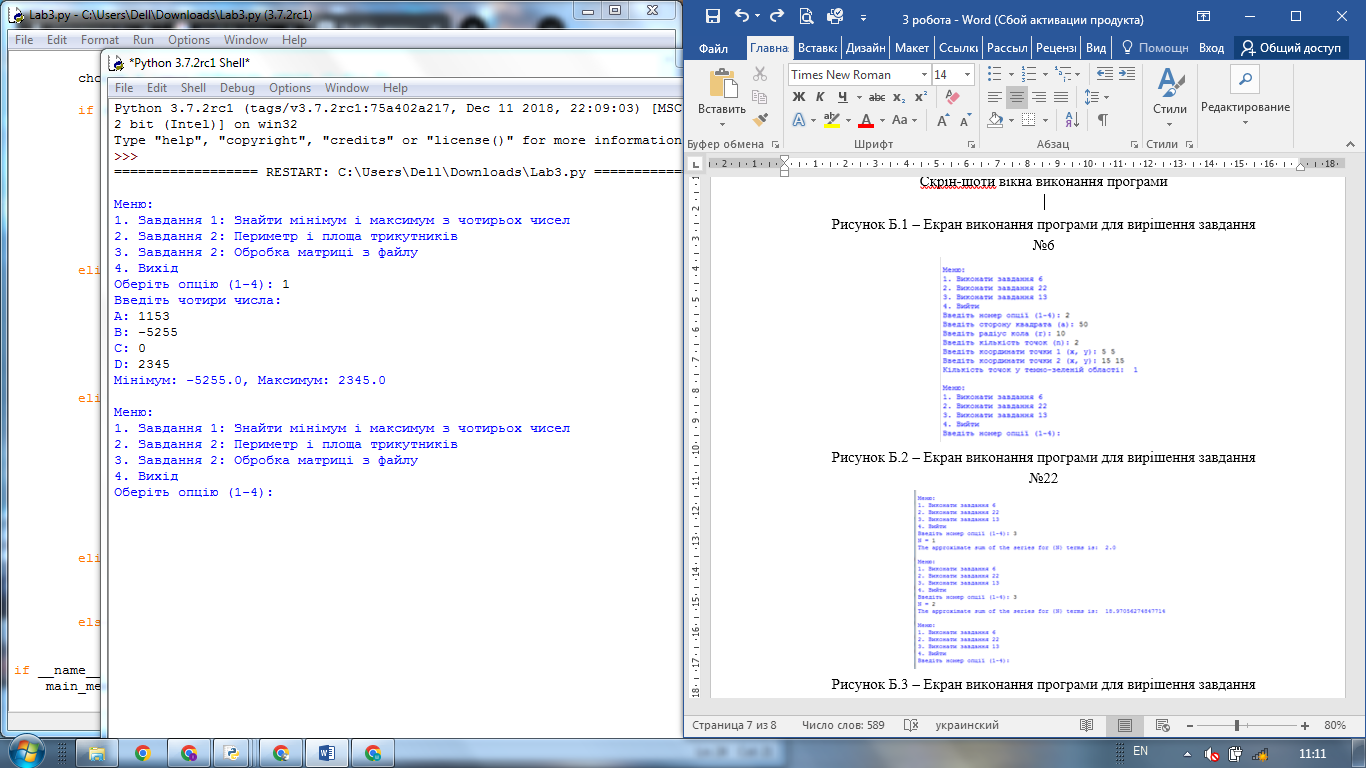


Рисунок Б.1 – Екран виконання програми для вирішення завдання   
№11

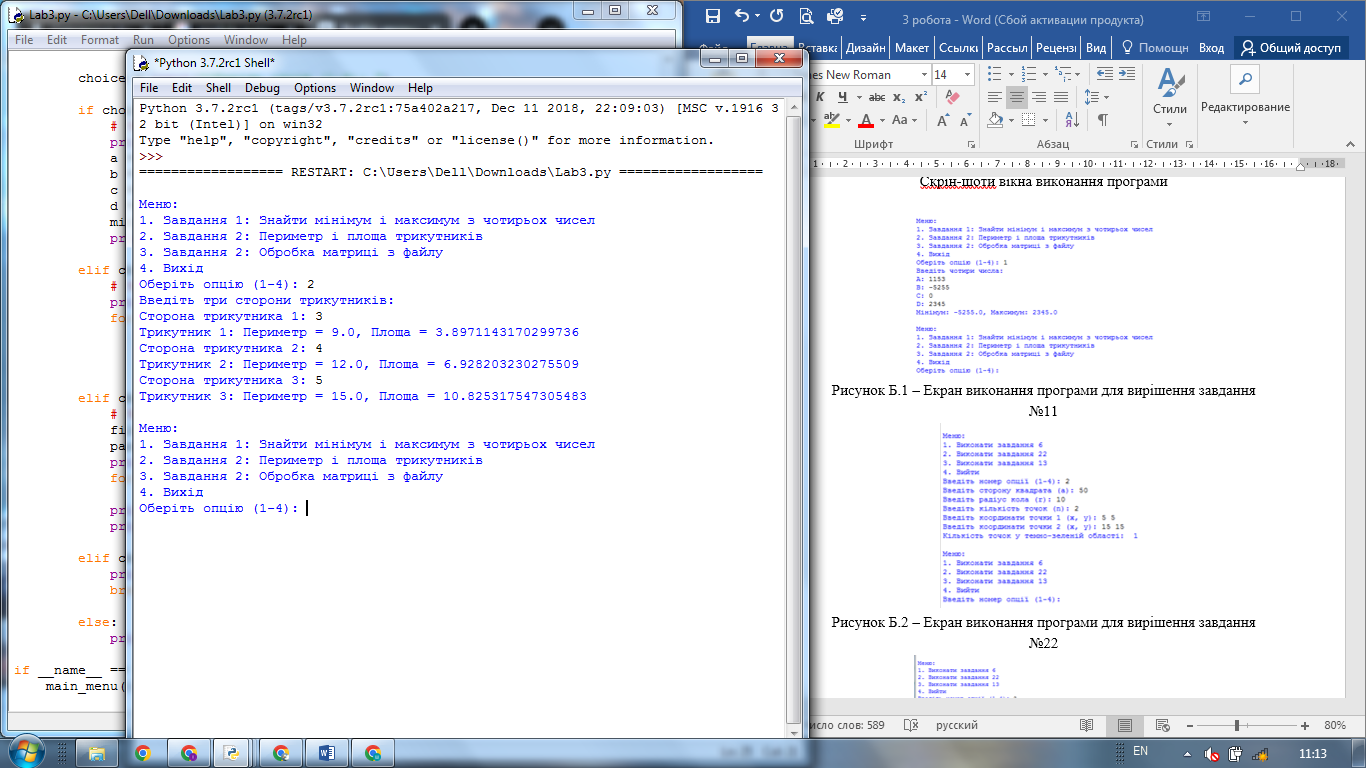


Рисунок Б.2 – Екран виконання програми для вирішення завдання   
№4

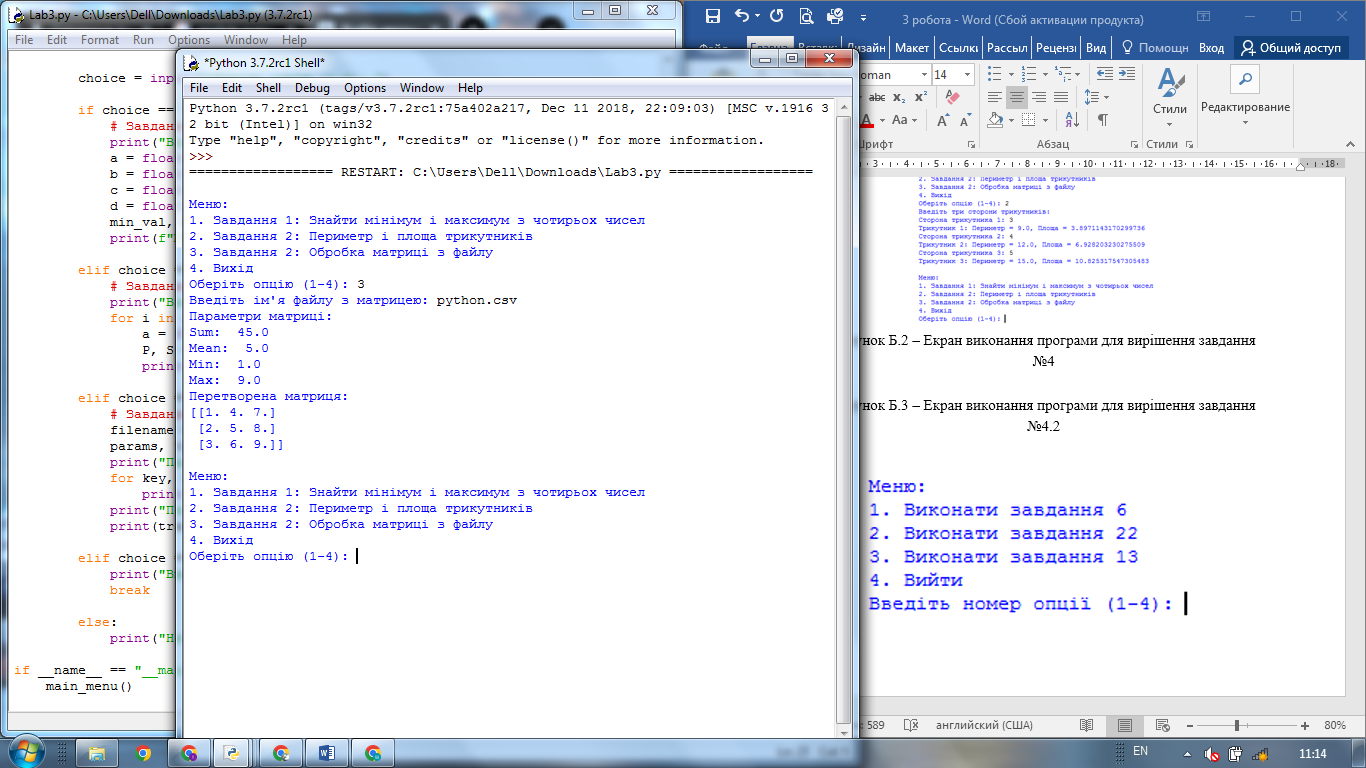


Рисунок Б.3 – Екран виконання програми для вирішення завдання   
№4.2