

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 _____»

Тема: «"Структурування програм з використанням
функцій"»

ХАІ.301.174.322.11 ЛР

Виконав студент гр. _____ 322 _____

_____ Шаєнко Віталій _____
(підпис, дата) (П.І.Б.)

Перевірив

_____ к.т.н., доц. О. В. Гавриленко
_____ ас. В. О. Білозерський
(підпис, дата) (П.І.Б.)

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. Вирішити завдання на алгоритми з розгалуженням.

Proc27. Описати функцію `DigitN (K, N)` цілого типу, яка повертає N -ю цифру ($0 < N \leq 4$) цілого додатного числа $0 < K < 1000$ (цифри в числі нумеруються справа наліво). Якщо кількість цифр в числі K менше N , то функція повертає -1 . Для кожного з п'яти даних цілих додатних чисел K_1, K_2, \dots, K_4 викликати функцію `DigitN` з параметром N , що змінюються від 1 до 4.

Завдання 2.

Matrix 5. У текстовому файлі задана матриця розміру $M \times N$. Для кожного рядка матриці з непарним номером (1, 3, ...) знайти середнє арифметичне і найменший з її елементів. Знайти скалярний добуток заданої матриці з матрицею того ж розміру, заповненої випадковими числами.

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Proc27.

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

K — змінна, загадане число, цілий тип;

Вихідні дані (ім'я, опис, тип):

result — змінна, список з п'яти чисел, цілий тип

Алгоритм вирішення показано нижче чи показано на рис. 1

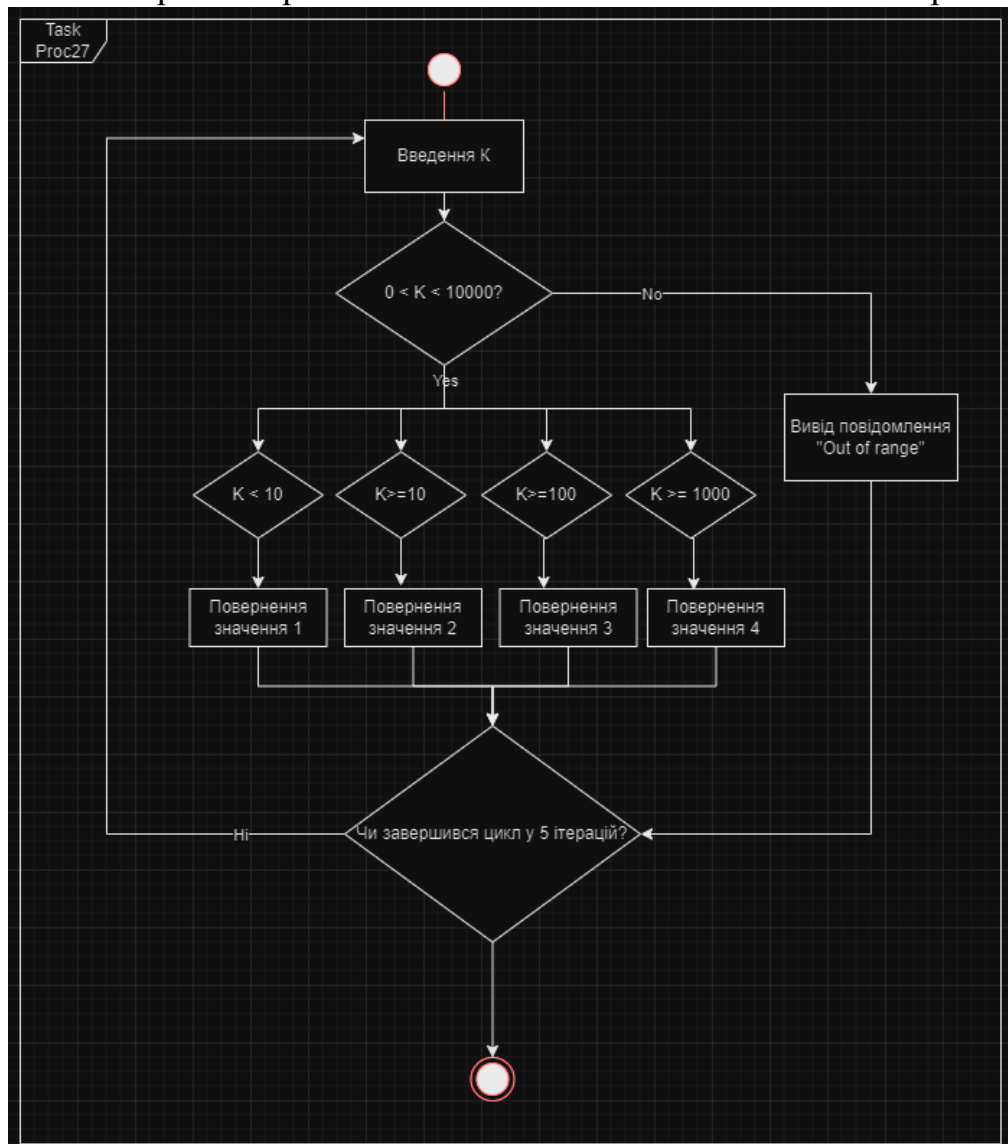


Рисунок 1 – Алгоритм вирішення Proc27

Лістинг коду вирішення задачі наведено в дод. А (стор. 6).

Екран роботи програми показаний на рис. Б.1.

Завдання 2. Вирішення задачі Matrix5.

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

M - int, розмір масиву в колонах

N - int, розмір масиву в стовпцях

Вихідні дані (ім'я, опис, тип):

scalar, min_values, summaries – float, скалярне число, мінімальні числа у кожному непарному стовпці масиву, середнє арифметичне кожного стовпцю масиву

Алгоритм вирішення <показано нижче чи показано на рис.2>

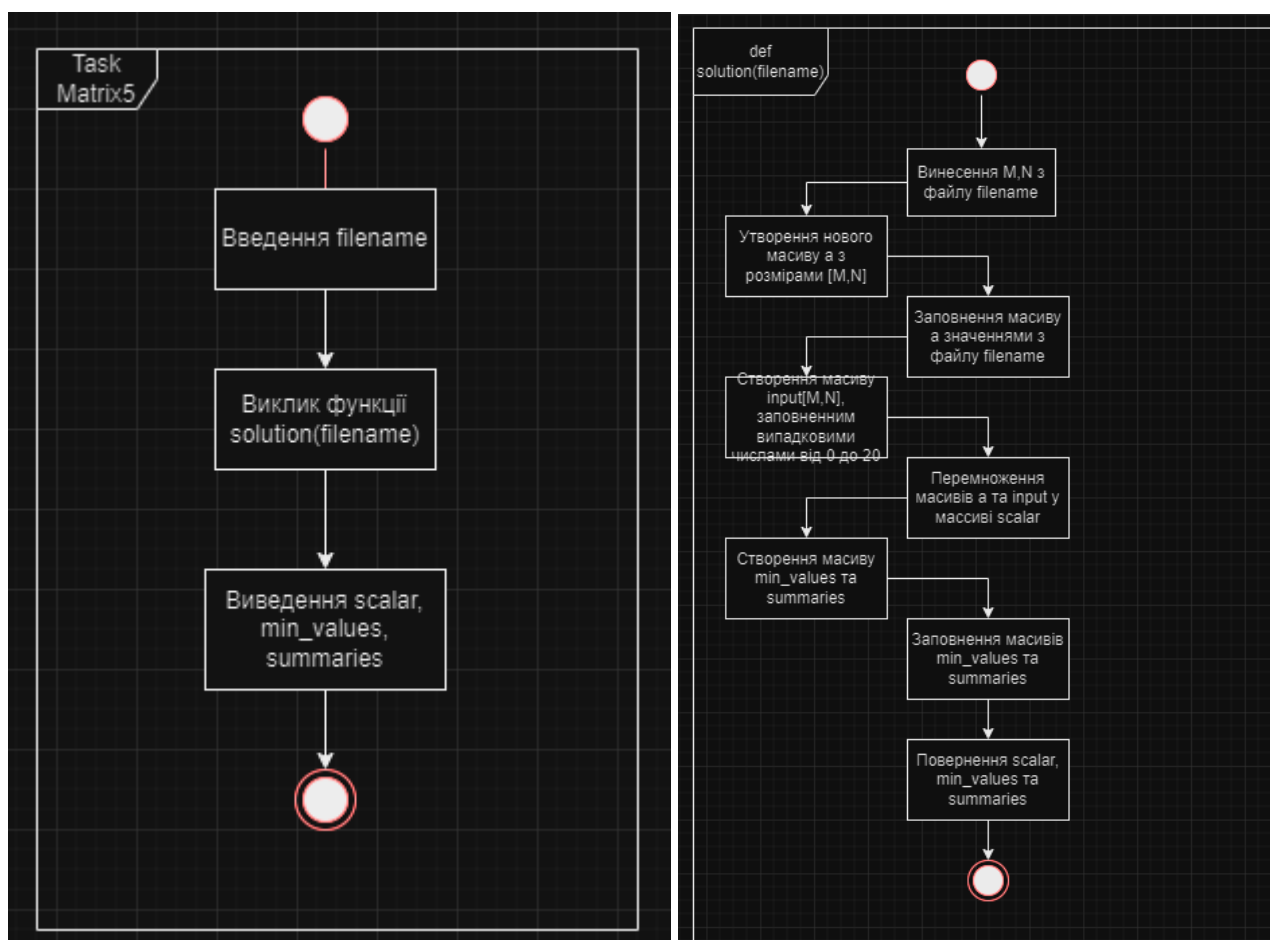


Рисунок 2 – Алгоритм вирішення Matrix5

Лістинг коду вирішення задачі наведено в дод. А (стор. 6).

Екран роботи програми показаний на рис. Б.2

ВИСНОВКИ

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

ДОДАТОК А.1

Лістинг коду основної програми запуску

```
import lab3

task_func_dictionary = {
    "1": lab3.task1,
    "2": lab3.task2,
}
if __name__ == '__main__':
    choice = input("Please, choose the task 1-2 (0-EXIT): ")
    while choice != "0":
        if choice in task_func_dictionary.keys():
            task_func_dictionary.get(choice)()
        else:
            print("Wrong task number!")
            choice = input("Please, choose the task again (0-EXIT): ")
```

ДОДАТОК А.2

Лістинг коду вирішення задач Proc27, Matrix5

```
import numpy as np
import random

def DigitCount(K):
    if 0 < K < 10000:
        if K >= 1000:
            return 4
        elif K >= 100:
            return 3
        elif K >= 10:
            return 2
        else:
            return 1
    else:
        return ("Out of range")

def task1():
    # Proc 27 (28 не було)
    try:
        arr = [int(input(f"({i + 1}) element :")) for i in range(5)]
        result = list(map(DigitCount, arr))
        print(result)

    except ValueError:
        print("Invalid input")

def solution(filename):
    with open(filename) as f:
        temp = f.readline().split(" ")
        M = int(temp[0])
        N = int(temp[1])

    input = np.loadtxt(filename, skiprows=1, max_rows=M)
    print(input)

    a = np.zeros((M, N))
```

```

for i in range (M):
    for j in range (N):
        a[i][j] = random.randint(0, 20)

multiply = a * input

min_values = []
summary = 0
summaries = []
for i in range(0, M, 2):
    min_value = min(input[i])
    min_values.append(min_value.item())
    for j in range(N):
        summary += input[i][j]

    summary /= N
    summaries.append(summary.item())
    summary = 0

return multiply, min_values, summaries

def task2():
    filename = ("input.txt")
    scalar, min_values, summaries = solution(filename)
    print("Умноженная матрица:", scalar)
    print("Минимальное число: ", min_values)
    print("Среднее арифметическое", summaries)

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
(1) element :  
450  
(2) element :  
210  
(3) element :  
5  
(4) element :  
1005  
(5) element :  
12  
[3, 3, 1, 4, 2]
```

Рисунок Б.1 – Екран виконання програми для вирішення завдання
Proc27

```
[[ 4.  5.  7.  3.]  
 [ 1.  2. 11. 12.]  
 [ 4. 13.  1.  7.]  
 [16. 10.  5.  4.]]  
Умноженная матрица: [[ 76.  60.  77.  6.]  
 [ 10.  38.  66.  72.]  
 [ 44. 195.  11.   0.]  
 [224. 180.  85.   8.]]  
Минимальное число: [3.0, 1.0]  
Среднее арифметическое [4.75, 6.25]
```

Рисунок Б.2 – Екран виконання програми для вирішення завдання
Matrix5

