НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра прикладної математики

Звіт

із лабораторної роботи №6

із дисципліни «Основи програмування»

на тему

РЕКУРСІЯ ТА ОБРОБКА МАТРИЦЬ

|  |  |
| --- | --- |
| Виконала: | Керівник: |
| студентка групи КМ-93 | ст.вик. Дрозденко О. М. |
| Довгаль Є. О. |  |

Київ — 2019

Зміст

[Зміст 2](#_Toc25685849)

[1 ПОСТАНОВКА ЗАДАЧІ 3](#_Toc25685850)

[1.1 Мета роботи 3](#_Toc25685851)

[1.2 Завдання до лабораторної роботи 3](#_Toc25685852)

[2 ОПИС ПРОГРАМИ 4](#_Toc25685853)

[2.1 Змінні та функції 4](#_Toc25685854)

[Завдання 1 4](#_Toc25685855)

[Завдання 2 5](#_Toc25685856)

[Додаток А 6](#_Toc25685857)

[Текст програми 6](#_Toc25685858)

[Додаток Б 8](#_Toc25685859)

[Скріншоти тестувань програми 8](#_Toc25685860)

# ПОСТАНОВКА ЗАДАЧІ

## Мета роботи

Вивчення роботи з одновимірними і двовимірними масивами із застосуванням рекурсивних функцій

## 1.2 Завдання до лабораторної роботи

1) Вивчити принципи і способи створення одновимірних і двовимірних масивів в мові Python.

2) Розробити програми відповідно до варіанта завдання.

3) Вхідні дані і результат роботи супроводжувати відповідною інформацією на екрані.

4) Показати розроблену програму викладачеві.

5) Письмово відповісти на Питання для самоперевірки.

6) Оформити звіт відповідно до вимог.

Завдання на виконання лабораторної роботи складається з 2-х частин:

− робота з одновимірними масивами (векторами); функції для обробки векторів повинні бути рекурсивними;

− робота з матрицями.

# 2 ОПИС ПРОГРАМИ

## Змінні та функції

### Завдання 1:

*Дан одновимірний масив числових значень, що нараховує n елементів. Поміняти місцями групу з m елементів, що починаються з позиції k з групою з m елементів, що починаються з позиції p.*

Пояснення:

Створено список, який буде зберігати наші числові значення – А.

Користувач задає через змінну n кількість елементів для А. Заповнюємо циклом for. Після отримуємо готовий для обробки список. Далі користувач задає три числа:

* s\_elems – кількість елементів перестановки (умова: 0 < s\_elems < n)
* first\_pos – позиція цих елементів (рахуємо з 1, так як ми – люди прості; умова: 0 < n-s\_elems+2 < n, тобто щоб не вийти за межу списка)
* second\_pos – позиція елементів, з якими міняємось місцями (вимоги аналогічні first\_pos)

Тепер сама обробка. Циклом for міняємо місцями елемент списка A[first\_pos] з А[second\_pos] зі зсувом вправо на кожній ітерації.

Готово!

### Завдання 2:

*Виконати обробку елементів прямокутної матриці A, що має n рядків і m стовпців. Знайти найменше значення серед середніх значень для кожного рядка матриці.*

Пояснення:

Для цього завдання я підключила бібліотеку NumPy (as np).

Створюємо два списки – А (майбутня матриця), means (середні значення по рядкам).

Користувач задає кількість стовпців (m) та рядків (n), після чого йде заповнення матриці циклом for, після чого вивід її користувачу.

Далі заповнюємо циклом for список means середніми значеннями (n разів через np.mean().

Нам потрібен найменше середнє значення, тому виводимо через min(means) його користувачеві, навіть номер рядка можемо назвати. Готово!

# Додаток А

## Текст програми

Зовнішній вигляд збережено для більш зручного читання

import numpy as np  
  
print('>>> Лабораторну роботу №6 виконала\n'  
 '>>> студентка КМ-93 Довгаль Єва\n'  
 '>>> Варіанти кожного завдання: 4\n')  
print('Суть завдання:\n'  
 ' 1) Дан одновимірний масив числових значень, що нараховує n елементів.\n'  
 ' Поміняти місцями групу з m елементів, що починаються з позиції k\n'  
 ' з групою з m елементів, що починаються з позиції p.\n'  
 ' 2) Виконати обробку елементів прямокутної матриці A, що має n рядків і m стовпців.\n'  
 ' Знайти найменше значення серед середніх значень для кожного рядка матриці\n')  
  
  
def get\_num(faq, param=0, gtype=int, expected=0):  
 errors = ('невід\'ємні', 'більші за 0', 'більші за 1')  
 type\_error = {int: 'цілі', float: 'дійсні'}  
 while True:  
 try:  
 num = gtype(input(f'{faq}: '))  
 if param and num < param-1:  
 print(f'⚠ Помилка! За умовою можна вводити лише {errors[param-1]} {type\_error[gtype]} числа.')  
 continue  
 if expected and not(0 < num < expected):  
 print(f'⚠ Помилка! За умовою можна вводити лише менші за {expected} {type\_error[gtype]} числа.')  
 continue  
 break  
 except ValueError:  
 print(f'⚠ Помилка! Вводити можна лише {type\_error[gtype]} числа.')  
 return num  
  
  
def task1():  
 A, n = [], get\_num('Введіть кількість елементів', 3)  
 for i in range(n):  
 A.append(get\_num(f'{i}', gtype=float))  
 print(f'>>> Отримано: {A}')  
  
 print('\n ▸ Кількість елементів перестановки < загальної кіль-ті елементів'  
 '\n ▸ Позиції елементів < кількість елементів перестановки\n')  
 s\_elems = get\_num('Введіть кількість елементів перестановки', expected=n)  
 first\_pos = get\_num('Введіть позицію цих елементів', expected=n-s\_elems+2)-1  
 second\_pos = get\_num('Введіть позицію елементів, з якими міняємося місцями', expected=n-s\_elems+2)-1  
 for i in range(s\_elems):  
 A[first\_pos+i], A[second\_pos+i] = A[second\_pos+i], A[first\_pos+i]  
 print(A)

def task2():  
 A, means = [], []  
 n = get\_num('Введіть кількість рядків', 2)  
 m = get\_num('Введіть кількість стовпців', 2)  
  
 for i in range(n):  
 A.append([])  
 for elem in range(m):  
 A[i].append(get\_num(f'{i + 1}.{elem + 1}', gtype=float))  
 print('\nОтримана матриця:\n', np.array(A))  
 for i in range(n): means.append(np.mean(A[i]))  
  
 print(f'Найменше середнє арифметичне на {means.index(min(means)) + 1} рядку: {min(means)}.')  
  
  
def restart\_task(task):  
 while True:  
 task()  
 restart = input('Продовжити? Y/N\n')  
 if restart.lower() != 'y': break  
  
  
while True:  
 choose\_task = input('Введіть номер завдання ("1" чи "2"): ').replace(' ', '')  
 if choose\_task == '1':  
 restart\_task(task1)  
 continue  
 if choose\_task == '2':  
 restart\_task(task2)  
 continue  
 if choose\_task.lower() == 'stop': break  
 print('Для завершення програми введіть "stop".')

# Додаток Б

## Скріншоти тестувань програми

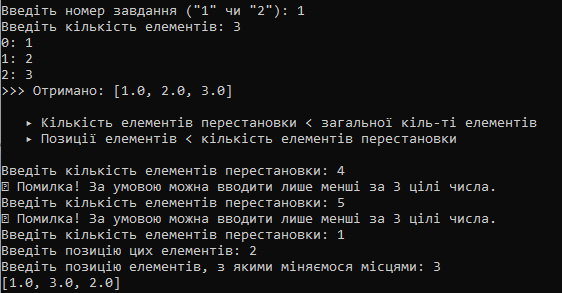


Рисунок 3.1 – Виконання завдання 1

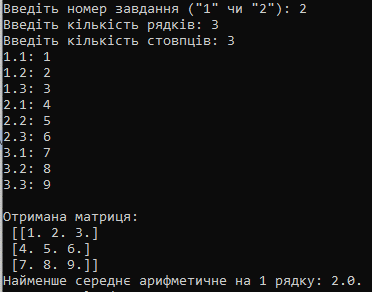


Рисунок 3.2 – Виконання завдання 2