МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Кафедра інформаційних технологій та програмування

**Звіт**

з лабораторної роботи №4

«Алгоритмічна структура розгалуження мовою Python»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Шарабар Ярослав Анатлійович

Перевірила:

викладач Устименко О.Б.

Оцінка\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ - 2023

**Зміст**

[Мета роботи 3](#_Toc150170352)

[1 Постановка задачі 4](#_Toc150170353)

[2 Основна частина 5](#_Toc150170354)

[2.1 Опис вхідних та вихідних даних 5](#_Toc150170355)

[2.1 Блок-схема 9](#_Toc150170356)

[Висновки 13](#_Toc150170357)

[Список літератури 15](#_Toc150170358)

[Додатки 16](#_Toc150170359)

# Мета роботи

Мета роботи полягає в ретельному вивченні складених типів даних у мові програмування Python, таких як списки, кортежі, словники та набори. Завдання включають вивчення основних властивостей кожного типу даних, роботу з методами та операціями, а також їхнє практичне застосування у розробці програм. Результатом має бути глибоке розуміння і ефективне використання складених типів даних для зберігання та обробки структурованої інформації в мові програмування Python.

# 1 Постановка задачі

**Варіант №20**

1. Використовуючи поняття списку, напишіть програму, яка створює 3D масив елементів a x b x c, кожен з яких має значення 0. Значення a, b, c вводяться в одному рядку через пропуск.
2. Напишіть програму, яка для двох послідовностей, що складаються з натуральних чисел, що не перевершують n, буде визначати, які числа зустрічаються в кожній з послідовностей, а які з чисел від 1 до n - в жодній з них. Спочатку на вхід програмі подається число n (1 ≤ n ≤ 255). У другому рядку вхідних даних знаходяться елементи першої послідовності, у третьому рядку - елементи другої послідовності. Виведіть у першому рядку в порядку зростання без повторень числа, які зустрічаються в кожній з послідовностей, а у другому рядку - в порядку зростання числа від 1 до 255, які не зустрічаються в жодній з них.

# 2 Основна частина

## 2.1 Опис вхідних та вихідних даних

**Задача № 1**

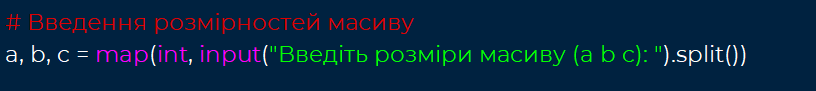
**Пояснення**

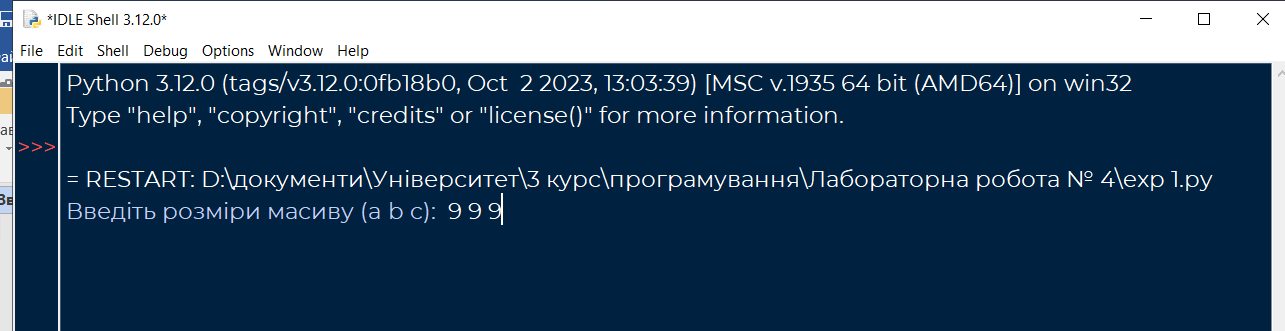
1. **Введення розмірностей масиву:**
   * Користувачеві пропонується ввести розміри масиву у вигляді трьох чисел, розділених пробілами (a, b, c).
   * **a**, **b**, і **c** - це цілі числа, які визначають розміри тривимірного масиву.
2. **Ініціалізація 3D масиву:**
   * Використовується генератор списку для створення тривимірного масиву (**three\_d\_array**) з розмірами **a x b x c**.
   * Кожен елемент масиву має значення 0.
3. **Виведення створеного масиву:**
   * Здійснюється виведення створеного тривимірного масиву.
   * Використовується вкладений цикл для ітерації по всіх елементах масиву та їх виведення в зручний формат.
   * Здійснюється перехід до нового рядка після кожного рядка та пустого рядка між "слоями" масиву.
4. **Оптимізація коду:**
   * Використано генератор списку замість вкладених циклів, що робить код більш читабельним та коротшим.
   * Використано розпакування **print(\*three\_d\_array[i][j])** для зручного виведення значень в рядок.

Ця програма виводить тривимірний масив із заданими розмірами, де кожен елемент масиву має значення 0. Вона демонструє використання списків та генераторів списку в мові програмування Python для створення та ініціалізації складених структур даних.

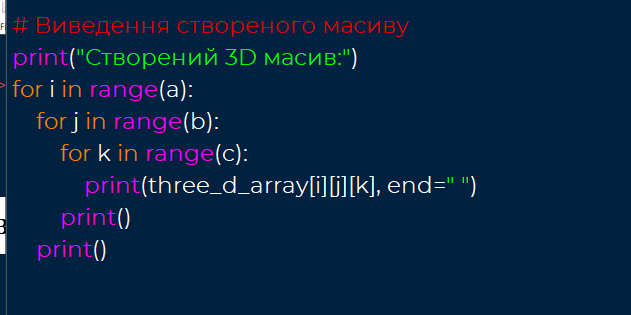
**Вхідні дані:**

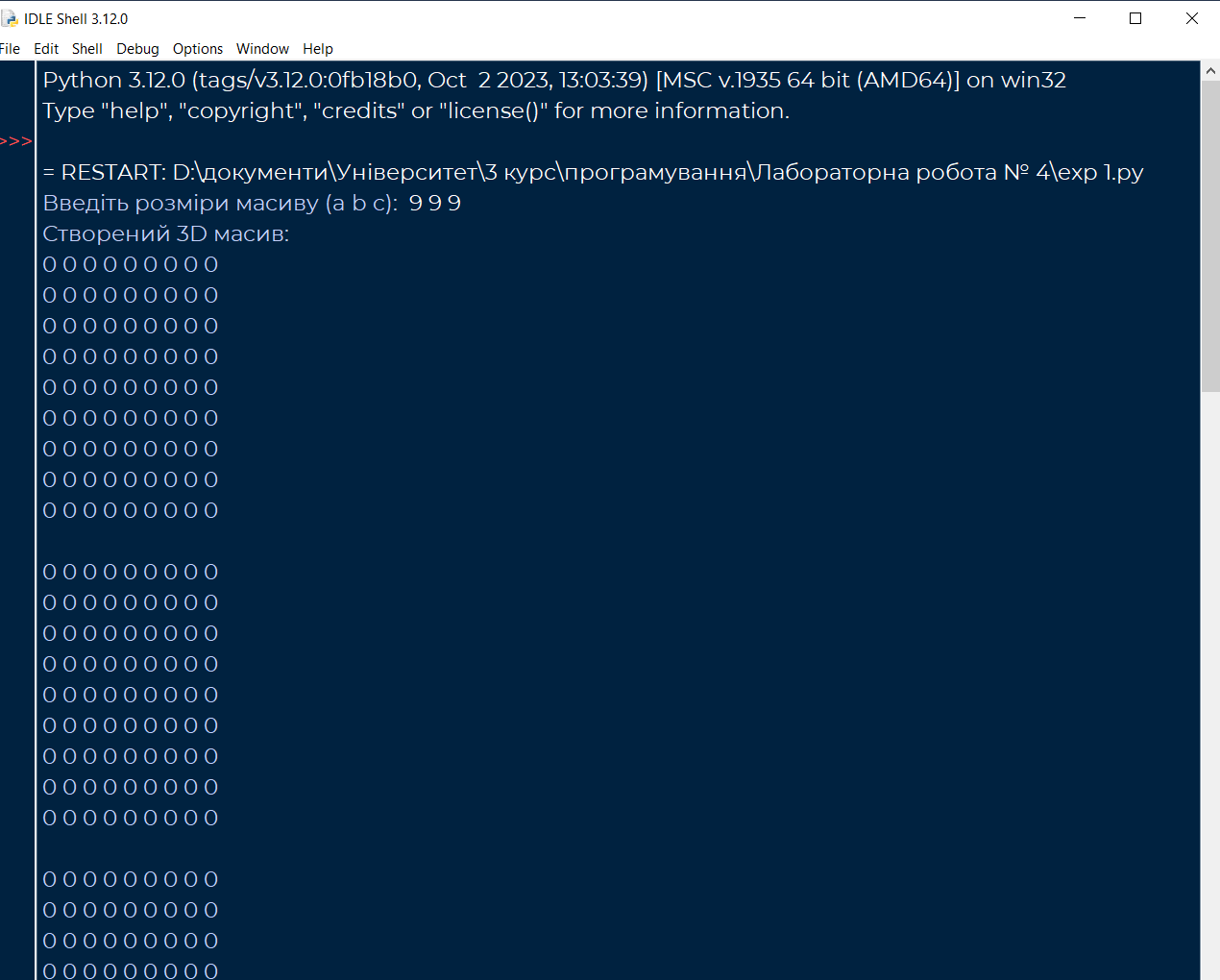
* Введення розмірностей масиву





**Вихідні дані:**





**Задача № 2**

n = int(input("Введіть число n (1 ≤ n ≤ 255): ")): Зчитує ціле число n від користувача.

seq1 = set(map(int, input("Введіть елементи першої послідовності: ").split())): Зчитує елементи першої послідовності від користувача, розділені пробілами, перетворює їх у цілі числа і зберігає у множині seq1.

seq2 = set(map(int, input("Введіть елементи другої послідовності: ").split())): Зчитує елементи другої послідовності від користувача, розділені пробілами, перетворює їх у цілі числа і зберігає у множині seq2.

common\_elements = sorted(seq1 & seq2): Знаходить спільні елементи двох послідовностей, сортує їх і зберігає у common\_elements.

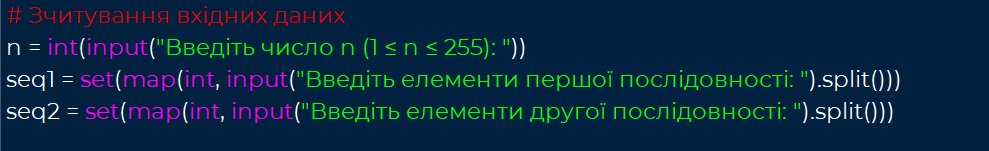
all\_elements = set(range(1, n + 1)): Створює множину всіх можливих елементів від 1 до n.

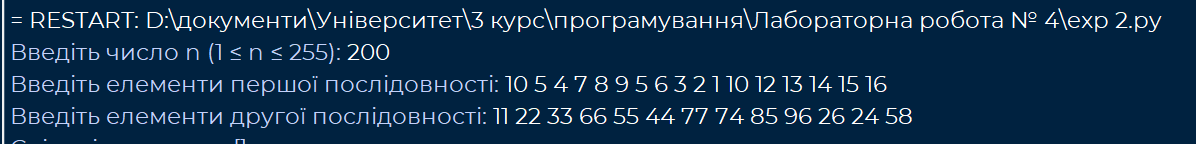
missing\_elements = sorted(all\_elements - (seq1 | seq2)): Знаходить елементи, які є у всіх можливих елементів, але не містяться в жодній з послідовностей, сортує їх і зберігає у missing\_elements.

print("Спільні елементи:", common\_elements): Виводить спільні елементи.

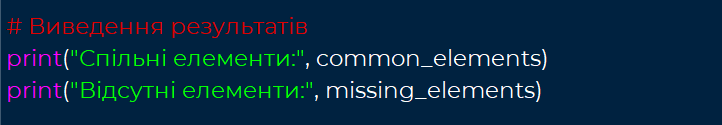
print("Відсутні елементи:", missing\_elements): Виводить елементи, які відсутні в обох послідовностях.

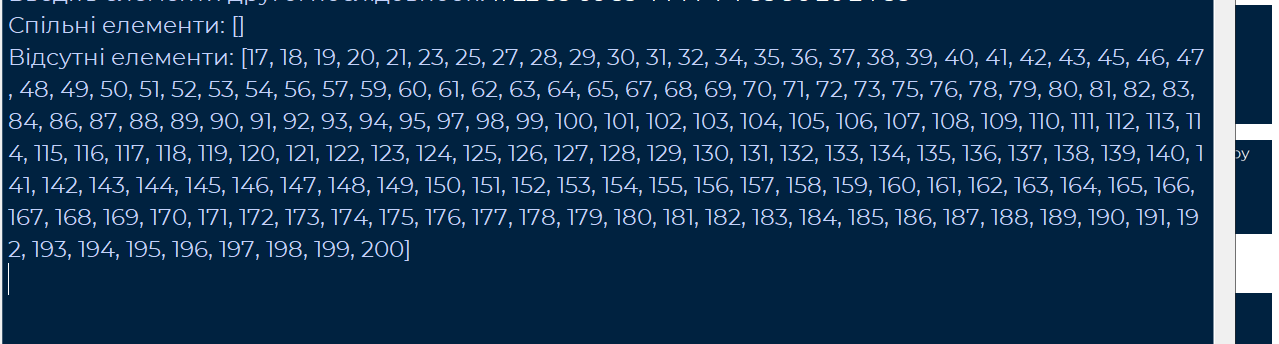
**Вхідні дані:**





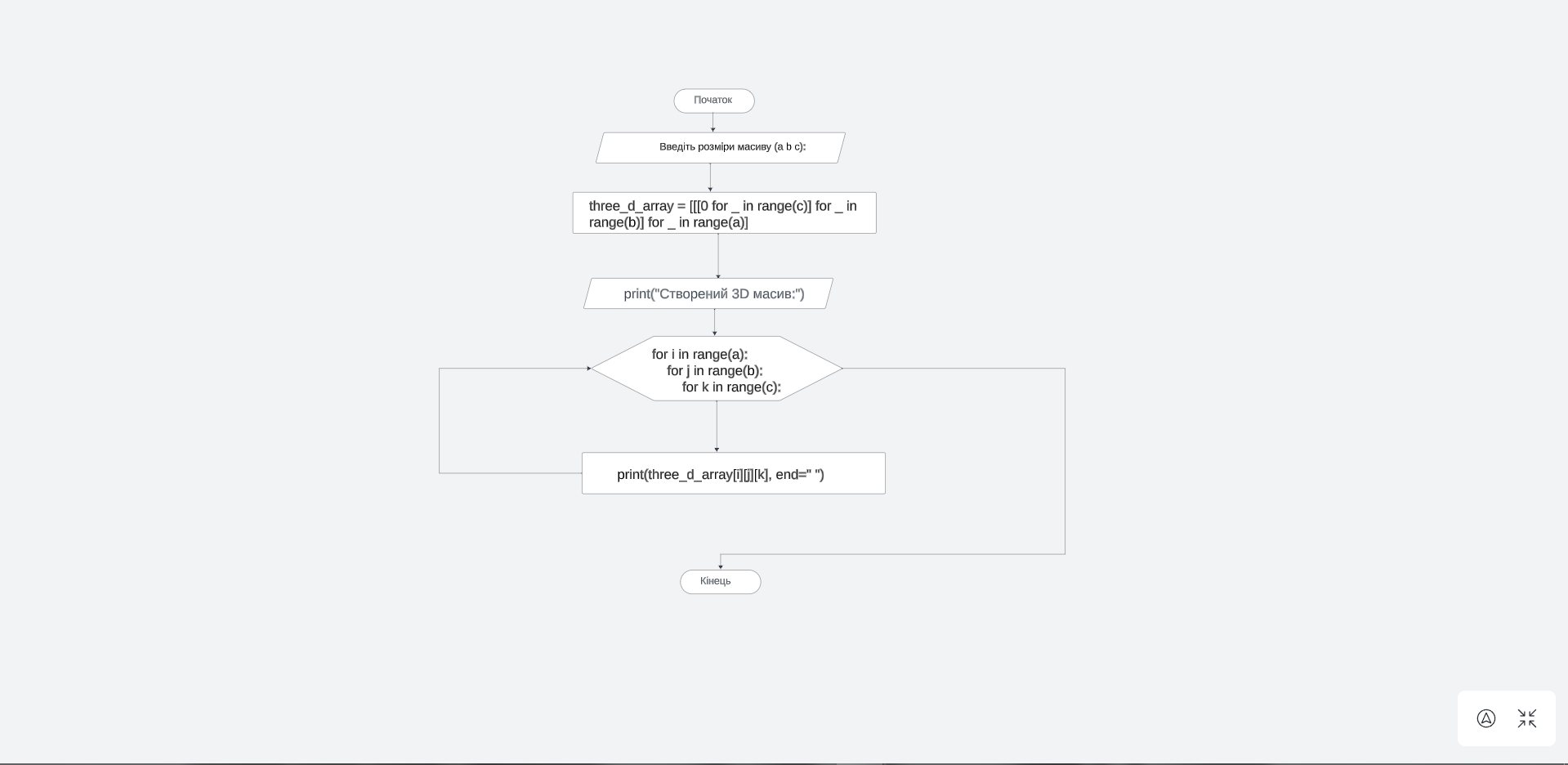
**Вихідні дані:**



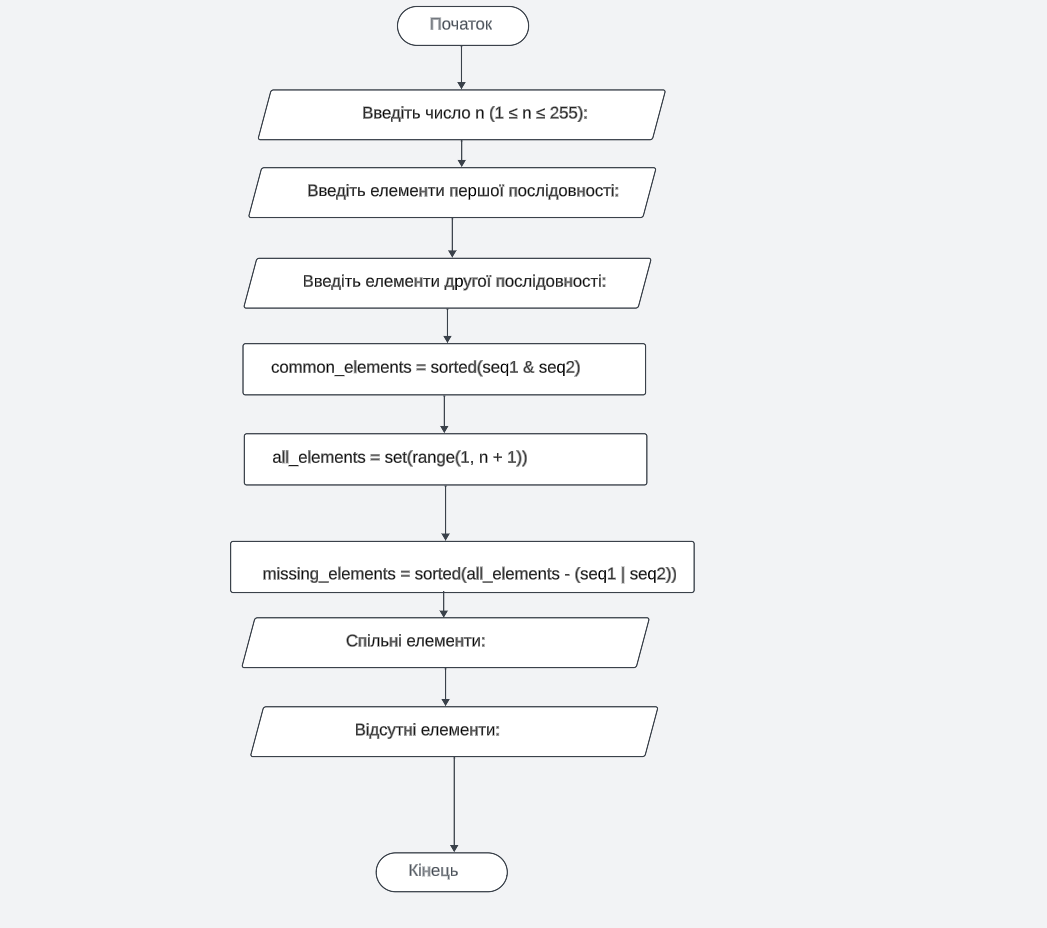


## 2.1 Блок-схема

**Задача №1**



**Задача №2**



# Висновки

Під час виконання лабораторної роботи ми детально вивчили складені типи даних у мові програмування Python та їх важливі характеристики. Виявилося, що кожен з цих типів (списки, кортежі, словники, набори) має свої унікальні властивості та призначення, що робить їх важливими інструментами для розв'язання різних завдань програмування. Списки дозволяють легко маніпулювати послідовностями елементів та надають широкий спектр методів для зручної роботи зі змінними даними. Кортежі, незмінні за своєю природою, стають відмінним вибором для зберігання сталих наборів даних. Словники, які працюють за принципом ключ-значення, відкривають можливості для швидкого доступу до даних за допомогою унікальних ідентифікаторів. Набори дозволяють зберігати унікальні елементи, що робить їх ефективним інструментом для роботи з унікальними значеннями.

Практичне використання складених типів даних в програмах дозволяє ефективно вирішувати різноманітні задачі, такі як сортування, фільтрація, ітерація, а також зберігання та обробка структурованої інформації. Отримані знання виявляться корисними при розробці програм з великою кількістю даних та складною логікою.

В цілому, лабораторна робота забезпечила нас глибоким розумінням та практичним досвідом роботи зі складеними типами даних у Python, що стане невід'ємною складовою навичок програміста.

# Список літератури

1. [Усі команди Python](https://foxminded.ua/python-vsi-komandy/)…………………………………………………..
2. [Мудл програмування](https://moodle.fmif.udu.edu.ua/course/view.php?id=939)…………………………………………………
3. [Блок схеми алгоритмів](https://yevshan.com.ua/info/006/content/content3.html)……………………………………………….

# Додатки

Лістинги програм

