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 №3

«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 лабораторної роботи "Цикли у Python" полягає в ознайомленні з основними конструкціями циклів у мові програмування Python, розвитку їх навичок роботи з циклами та вміння використовувати їх для розв'язання різноманітних завдань. Це включає в себе вивчення роботи з циклами for та while, роботу зі списками, умовні оператори всередині циклів, використання функцій, обробку рядків та вирішення завдань на основі реальних сценаріїв. Лабораторна робота також спрямована на розвиток алгоритмічного мислення та практичних навичок розв'язання завдань за допомогою циклів у програмуванні.

# 1 Постановка задачі

**Варіант №21**

1. Напишіть програму, яка виводить двовимірну квадратну таблицю n x n чисел (n - ціле число, яке вводить користувач). По головній діагоналі таблиці (від лівого верхнього значення до нижнього правого значення) мають розміщуватися 0, вище головної діагоналі розміщуються 1, а нижче - -1. Для гарного відображення таблиці при виведенні її значень використовуйте символ табуляції \t.

2. Дано ціле число n. З чисел 1, 4, 9, 16, 25, … надрукувати ті, які не перевищують **n.**

# 2 Основна частина

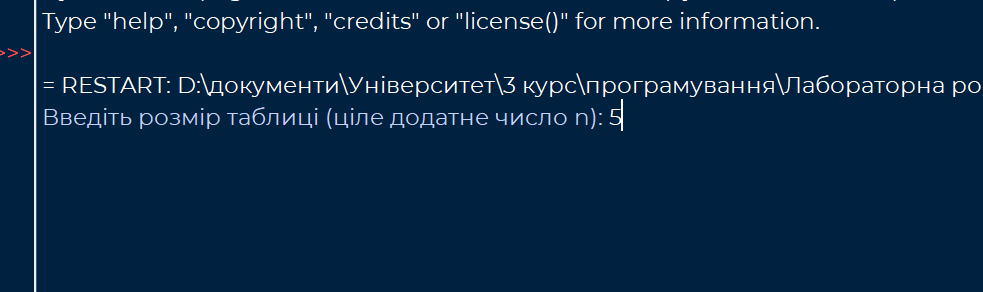
## 2.1 Опис вхідних та вихідних даних

**Задача № 1**

**Пояснення**

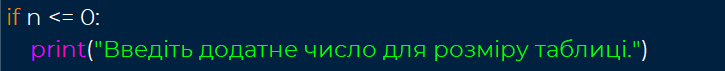
1 Введення розміру таблиці:





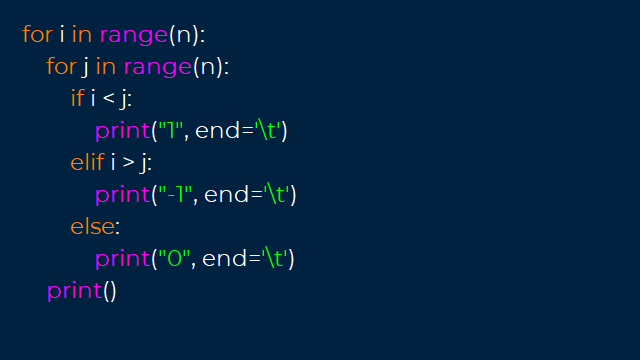
користувач вводить ціле число n, яке визначає розмір квадратної таблиці.

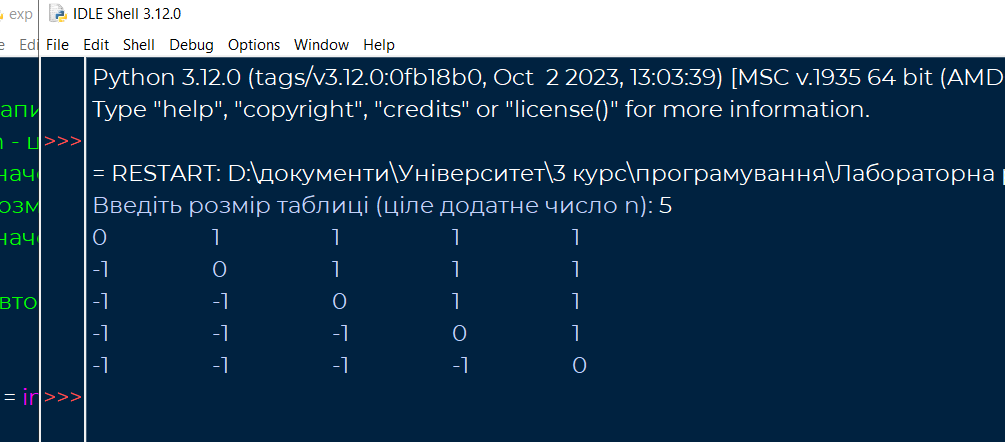
2 Перевірка на додатність числа n:



Це умовна конструкція, яка перевіряє, чи n менше або дорівнює 0. Якщо це так, виводиться повідомлення про те, що потрібно ввести додатне число, і програма переходить далі.

3 Виведення таблиці:





Це вкладений цикл for, який генерує та виводить квадратну таблицю розміром n x n. Умовні конструкції if-elif-else визначають значення кожної комірки таблиці відповідно до вказаних у завданні правил.

Якщо i менше j, виводиться 1 (вище головної діагоналі).

Якщо i більше j, виводиться -1 (нижче головної діагоналі).

Якщо i дорівнює j, виводиться 0 (по головній діагоналі).

Кожен рядок таблиці завершується символом нового рядка print() для оформлення таблиці.

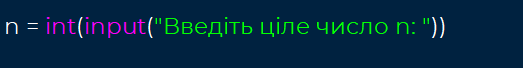
**Задача № 2**

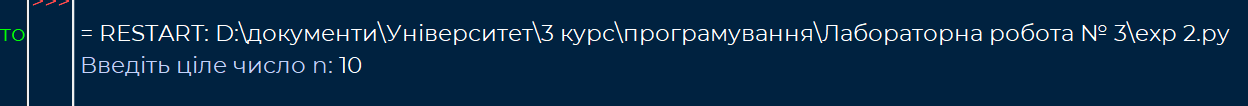
Звідси почнемо ітерацію з i = 1 та змінної square = 1. У кожній ітерації ми друкуємо поточне значення square, а потім збільшуємо i на 1 і обчислюємо нове значення square як квадрат поточного i (тобто, square = i\*\*2).

Цикл буде тривати до тих пір, поки square не перевищить введене число n. Коли square стає більшим або рівним n, цикл завершується, і програма завершує роботу. Таким чином, програма друкує квадрати чисел 1, 4, 9, 16, і т.д., які не перевищують введене число n.

**Вхідні дані:**

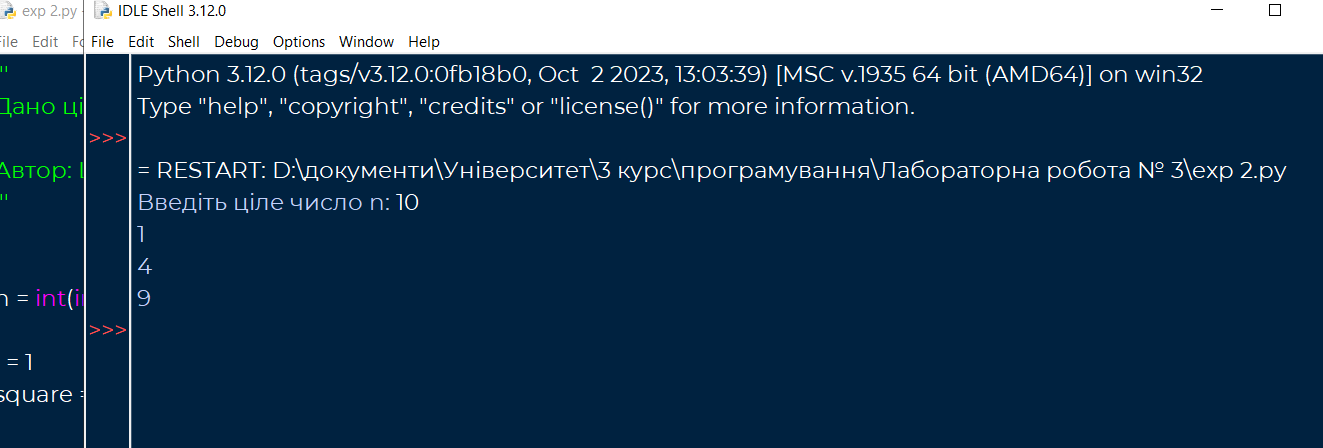
* Введіть ціле число n.





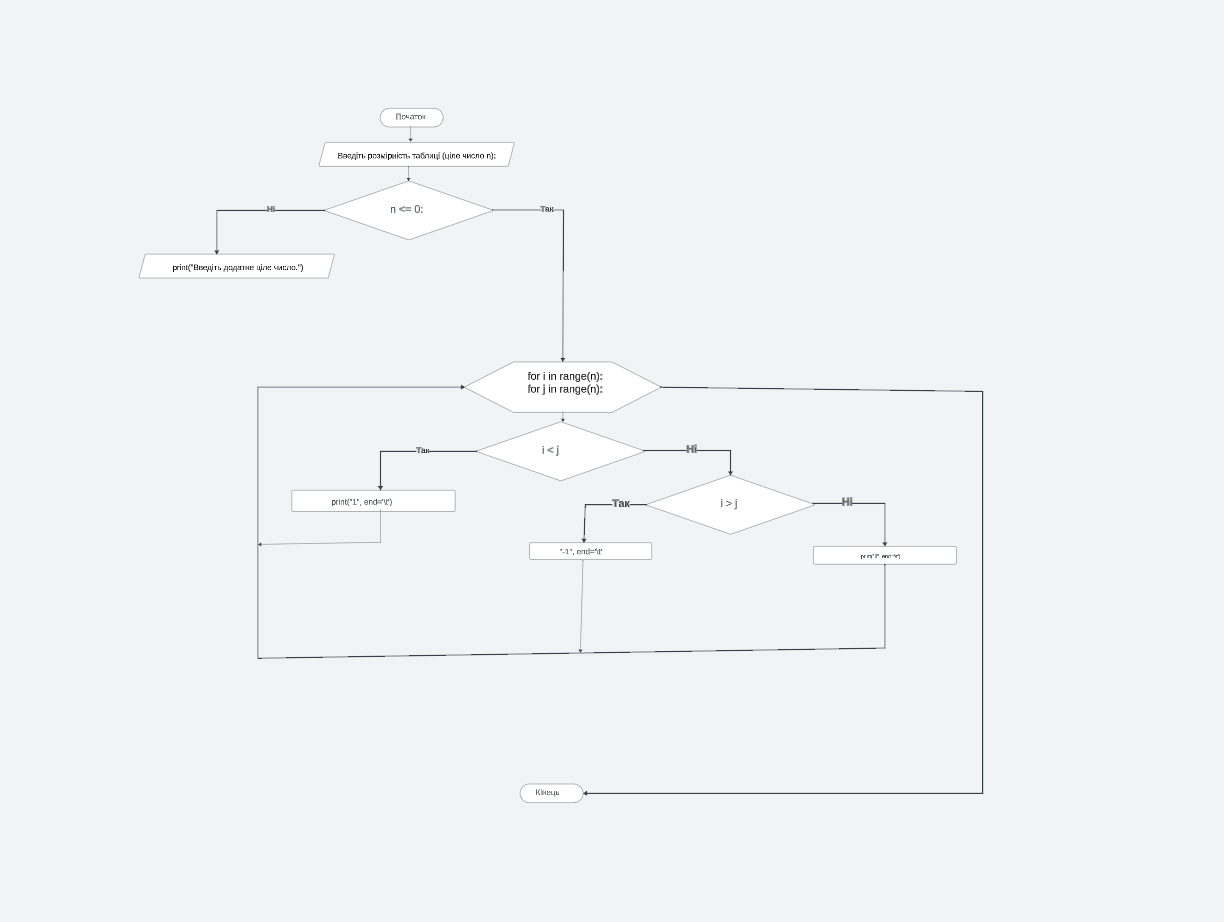
**Вихідні дані:**



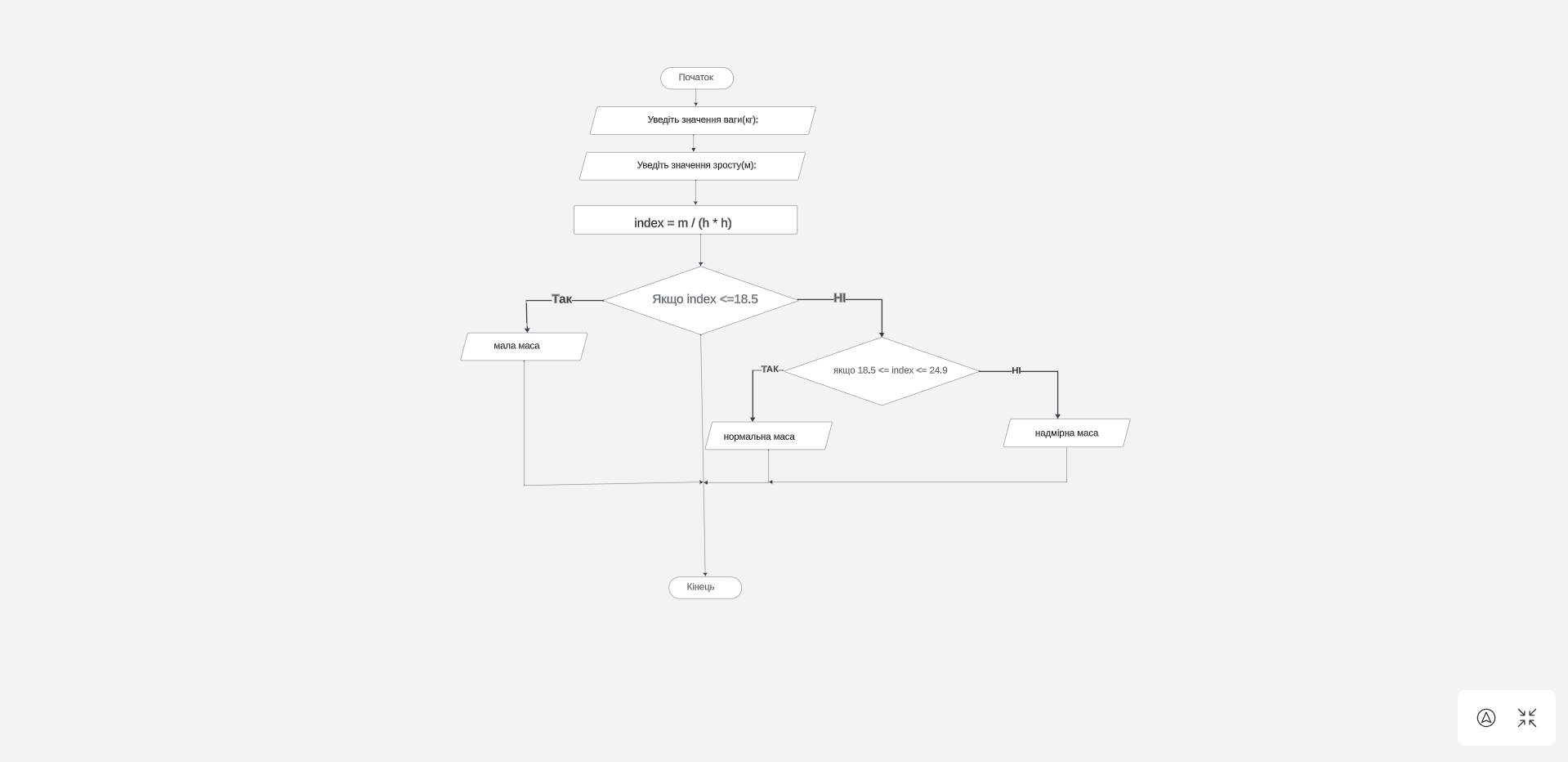


## 2.1 Блок-схема

**Задача №1**



**Задача №2**



# Висновки

У результаті виконання лабораторної роботи було досягнуто кілька ключових висновків:

* Освоєння алгоритмічної структури розгалуження:

Студенти ознайомилися з використанням умовних операторів (if, elif, else) в мові програмування Python.

Вивчення цих конструкцій дозволяє студентам створювати програми, які приймають рішення на основі умов.

* Навчання використанню логічних операцій:Використання логічних операцій (and, or, not) в контексті умовних операторів дозволяє структурувати та оптимізувати умови в програмах.
* Практичні навички в програмуванні:Завдання, пов'язані з розрахунком індексу маси тіла та перевіркою унікальності цифр у числі, дозволяють студентам застосовувати здобуті знання для вирішення конкретних завдань.
* Застосування умов та виведення результатів:Програми, які використовують умовні оператори, можуть адаптуватися до різних сценаріїв в залежності від введених даних.Виведення результатів роботи програм дозволяє користувачам отримати інформацію або зрозуміти, як програма обробляє вхідні дані.

Загалом, лабораторна робота стала важливим етапом у формуванні навичок у програмуванні, роботі з умовами та логічними операціями, що є фундаментальними для подальшого вивчення програмування.

# 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

