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 №2

«Алгоритмічна структура розгалуження мовою Python»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Гладкович Я.В

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 7](#_Toc150170356)

[Висновки 10](#_Toc150170357)

[Список літератури 12](#_Toc150170358)

[Додатки 13](#_Toc150170359)

# Мета роботи

Мета лабораторної роботи – скласти програми мовою Python для вирішення задач з теми «Алгоритмічна структура розгалуження мовою Python»

# 1 Постановка задачі

1. Напишіть програму, яка запропонує користувачеві ввести десяткове число у межах від 1 до 10. Програма повинна вивести версію римського числа. Програма має враховувати ситуацію, якщо введене число є за межами діапазону від 1 до 10.
2. Шаховий король ходить по горизонталі, вертикалі і діагоналі, але тільки на 1 клітинку. Дано дві різні клітинки шахової дошки, визначте, чи може король потрапити з першої клітинки на другу за один хід. Програма отримує на вхід чотири числа від 1 до 8 кожне, що задають номер стовпця і номер рядка спочатку для першої клітинки, потім для другої клітинки. Програма повинна вивести Yes, якщо з першої клітинки ходом короля можна потрапити в другу або No в іншому випадку.

# 

# 2 Основна частина

## 2.1 Опис вхідних та вихідних даних

***Завдання №1***

Команди, використані для цієї програми:

**1. `print()`:** Вивід тексту на екран.

**2. `input()`:** Запит користувача ввести число з клавіатури.

**3. `int()`:** Перетворення введеного користувачем рядка на ціле число.

**4. `if` та `else`:** Умовні конструкції для визначення, чи введене число знаходиться у межах від 1 до 10.

**5. Функція `decimal\_to\_roman(n)`:** Декларація та визначення функції для перетворення десяткового числа у римське.

**6. `return`:** Повернення значення з функції.

**7. Створення та використання словника (`roman\_numerals`)** для відображення відповідних римських чисел.

**1. Виведення текстового повідомлення:**

print("Введіть число від 1 до 10")

Повідомлення виводиться на екран для користувача з проханням ввести число від 1 до 10.

**2. Зчитування введеного числа від користувача:**

number = int(input("Введіть число: "))

Користувач вводить десяткове число, яке потім перетворюється у ціле.

**3. Функція для перетворення в римське число:**

def decimal\_to\_roman(n):

if n < 1 or n > 10:

return "Помилка."

else:

roman\_numerals = {1: 'I', 2: 'II', 3: 'III', 4: 'IV', 5: 'V', 6: 'VI', 7: 'VII', 8: 'VIII', 9: 'IX', 10: 'X'}

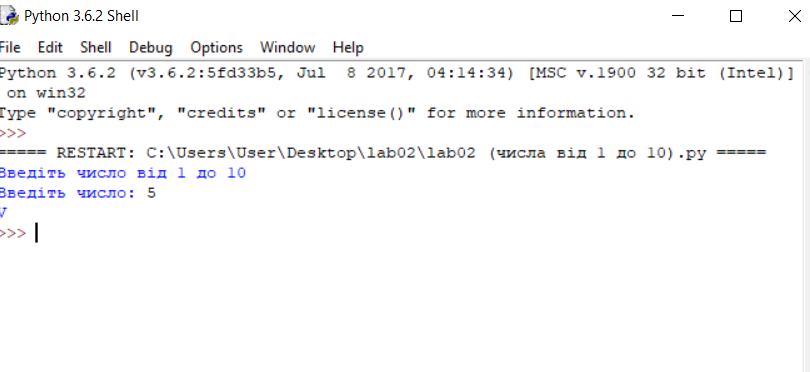
return roman\_numerals[n]

Функція приймає десяткове число `n` і перевіряє, чи воно знаходиться у межах від 1 до 10. Якщо так, функція повертає відповідне римське число; в іншому випадку повертається повідомлення про помилку.

**4. Виведення результату перетворення:**

print(decimal\_to\_roman(number))

Виводиться результат роботи функції для введеного користувачем десяткового числа. Якщо число не входить у діапазон від 1 до 10, виводиться повідомлення про помилку.



***Завдання №2***

**Команди, використані для цієї програми:**

**1. `def can\_king\_reach(x1, y1, x2, y2)`:** Оголошення та визначення функції для визначення, чи може король потрапити з однієї клітинки на іншу за один хід.

**2. `abs()`:** Функція для знаходження абсолютного значення числа.

**3. `return`:** Повернення значення з функції.

**4. `map()`:** Застосування функції `int` до кожного елемента ітерабельного об'єкта.

**5. `input()`:** Зчитування введених користувачем даних з клавіатури.

**6. `split()`:** Розбиття рядка на список за пробілами.

**1. Оголошення та визначення функції:**

def can\_king\_reach(x1, y1, x2, y2):

if abs(x2 - x1) <= 1 and abs(y2 - y1) <= 1:

return "Так"

else:

return "Ні"

Функція приймає координати двох клітинок (`x1`, `y1`, `x2`, `y2`) та визначає, чи може король потрапити з першої клітинки на другу за один хід. Вона порівнює різницю між x- та y-координатами та перевіряє, чи вона менше або дорівнює 1.

**2. Введення координат від користувача:**

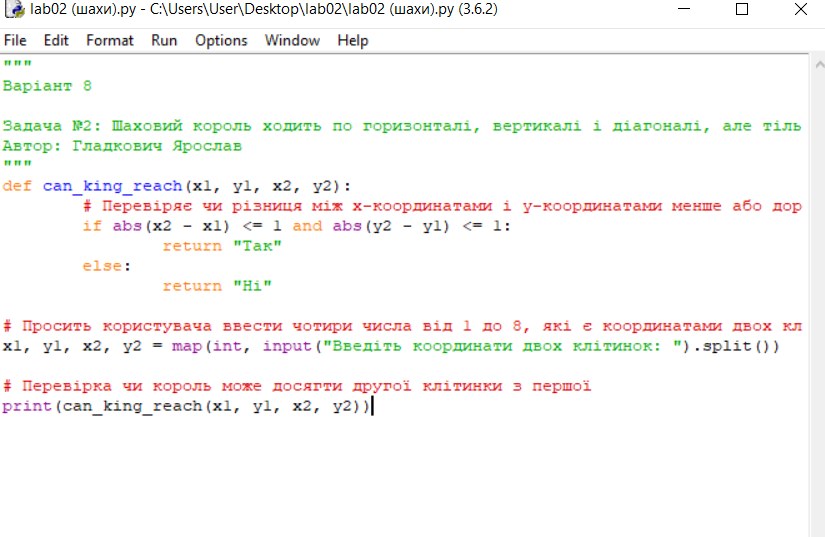
x1, y1, x2, y2 = map(int, input("Введіть координати двох клітинок: ").split())

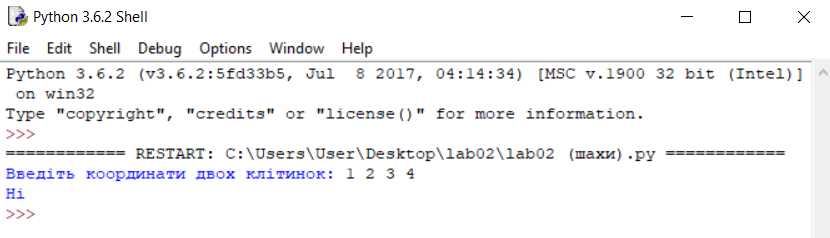
Користувач вводить чотири числа, які представляють координати двох клітинок.

**3. Перевірка можливості досягнення другої клітинки з першої:**

print(can\_king\_reach(x1, y1, x2, y2)

Викликається функція `can\_king\_reach` з введеними користувачем координатами, і результат виводиться на екран.





## 2.1 Блок-схема завдання №1

Початок

print("Введіть число від 1 до 10")

Кінець

return roman\_numerals[n]

def decimal\_to\_roman(n): if n < 1 or n > 10: return "Помилка." else: roman\_numerals = {1: 'I', 2: 'II', 3: 'III', 4: 'IV', 5: 'V', 6: 'VI', 7: 'VII', 8: 'VIII', 9: 'IX', 10: 'X'}

number = int(input("Введіть число: ")))

**Блок-схема завдання №2**

Початок

x1, y1, x2, y2 = map(int, input("Введіть координати двох клітинок: ").split())

Кінець

print(can\_king\_reach(x1, y1, x2, y2)

def can\_king\_reach(x1, y1, x2, y2): if abs(x2 - x1) <= 1 and abs(y2 - y1) <= 1: return "Так" else: return "Ні"

# Висновки

У ході виконаної лабораторної роботи з теми "Алгоритмічна структура розгалуження мовою Python" були розглянуті різноманітні завдання, спрямовані на використання умовних конструкцій та алгоритмів в мові програмування Python. Програми були написані для розв'язання різних практичних завдань, що вимагали обробки умов та прийняття відповідних рішень.Кожна програма включала в себе введення даних, обробку цих даних за допомогою різних умовних конструкцій, та виведення результату. Важливим аспектом було правильне розуміння умов задачі та ефективне використання алгоритмів для досягнення необхідного результату.Основні концепції, які були використані у програмах, включають умовні конструкції (if-else), логічні операції (and, or), а також роботу з введеними даними та виведенням результату на екран. Кожен варіант завдання вимагав уважного аналізу та використання правильного алгоритму для досягнення бажаного результату.

Лабораторна робота допомогла зрозуміти важливість алгоритмічних рішень та структури розгалуження в розв'язанні конкретних задач. Отримані навички роботи з умовами та логікою програми є важливим етапом у вивченні алгоритмів та програмування в цілому.

# Список літератури

Список літератури

# Додатки

https://github.com/YH31I/Lab02