**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування на мові Python**

Звіт до Теми №1

Функції та змінні

Під час виконання практичного завдання до Теми №1 було надано варіанти рішення до наступних задач:

**Перетворення рядка**

Необхідно рядок, що має вигляд "abcdefg123" перетворити наступним чином "321gfedcba", вважаючи сталою довжину рядку в 10 символів.

Хід виконання завдання:

В ПЗ пишем код:

Крок 1: Створюємо функцію 'revers', яка містить рядок 's' зі значенням "abcdefg123"

Крок 2: Перевертаємо рядок 's' за допомогою зрізу '[::-1]' і зберігаємо результат у змінну 'result'. (Перший порожній елемент означає "почни з першого символу" (або з кінця, бо ми йдемо у зворотному напрямку). Другий порожній елемент означає "йди до останнього символу" та -1 вказує на те, що ми рухаємося у зворотному порядку.

Крок 3: Функція повертає перевернутий рядок, який зберігається в 'result'

Текст програми:

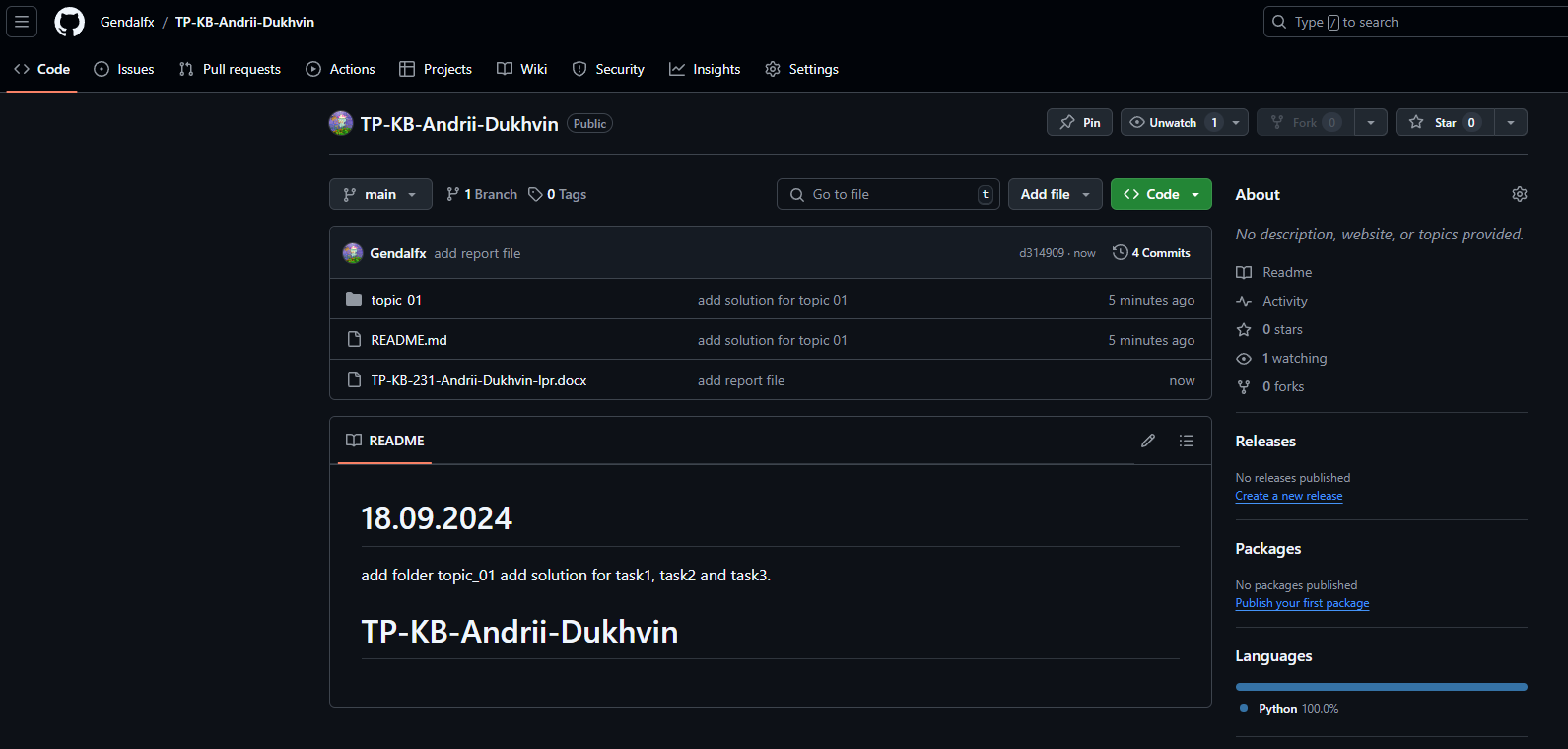
|  |
| --- |
| def revers():      s ="abcdefg123"      result = s[::-1]      return result  print (revers()) |

При компіляції він виводить:



Посилання на github: https://github.com/Gendalfx/TP-KB-Andrii-Dukhvin

Знімок екрану з посилання на github:



**Виконати тестування функцій, що працюють з рядками: strip(), capitalize(), title(), upper(), lower().**

Текст програми:

def test\_string\_functions():

    s = "   hello world! welcome to python.   "

    # strip() - Видаляє пробіли з обох боків рядка

    stripped = s.strip()

    print(f"strip():'{stripped}'")

   # capitalize() - Робить першу літеру великою, а інші маленькими

    capitalized = stripped.capitalize()

    print(f"capitalize():'{capitalized}'")

    # title() - Робить першу літеру кожного слова великою

    titled = s.title()

    print(f"title():'{titled}'")

    # upper() - Перетворює всі символи в рядку на великі

    uppercased = s.upper()

    print(f"upper():'{uppercased}'")

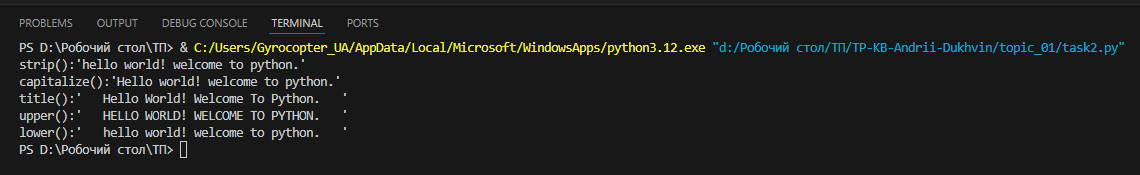
    # lower() - Перетворює всі символи в рядку на маленькі

    lowercased = s.lower()

    print(f"lower():'{lowercased}'")

test\_string\_functions()

При компіляції він виводить:



**Написати функцію пошуку дискримінанту квадратного рівняння.**

Формула дискримінанту: D=b^2−4ac

Текст програми:

def find\_discriminant(a, b, c):

    # Обчислюємо дискримінант

    discriminant = b\*\*2 - 4\*a\*c

    return discriminant

# Приклад використання

a = float(input("Введіть коефіцієнт a: "))

b = float(input("Введіть коефіцієнт b: "))

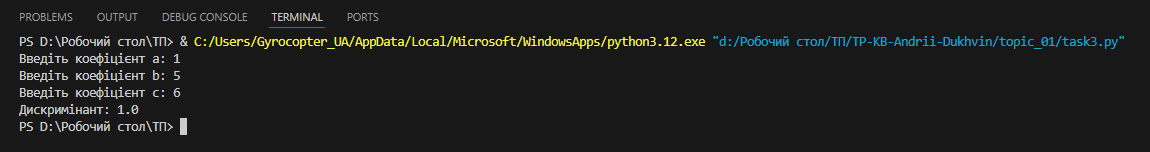
c = float(input("Введіть коефіцієнт c: "))

# Виклик функції та виведення результату

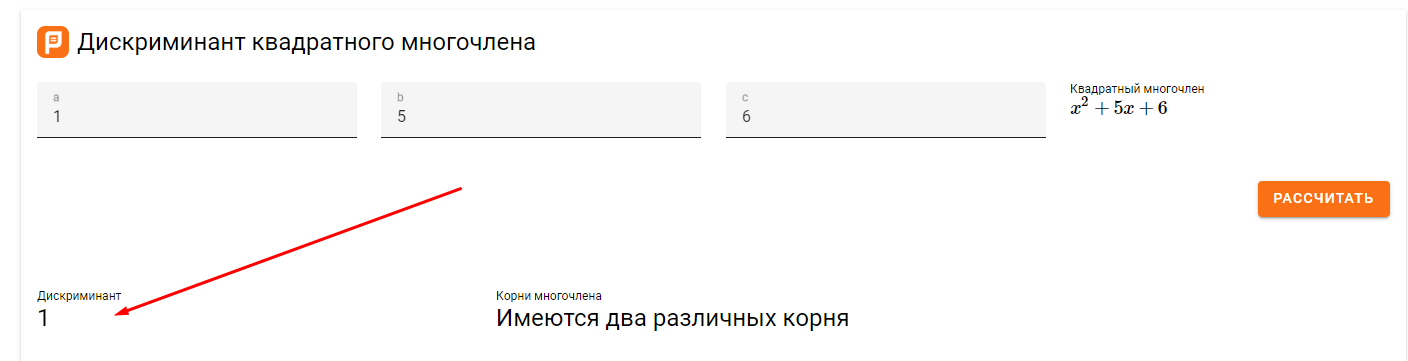
D = find\_discriminant(a, b, c)

print(f"Дискримінант: {D}")

При компіляції він виводить:



Перевірка:



Звіт до Теми №2

Умовний перехід

Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач:

**Завдання 1**

Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Хід роботи

**Визначення задачі**:

Програма повинна розв’язувати квадратні рівняння у формі ax^2+bx+c=0.

**Формули**:

* Обчислення дискримінанта: D=b2−4acD.
* Визначення коренів на основі значення дискримінанта:
  + Два дійсних корені, якщо D>0.
  + Один дійсний корінь, якщо D=0.
  + Два комплексних корені, якщо D<0.

**Написання функцій**:

* **Функція для обчислення дискримінанта**: приймає aaa, bbb, ccc і повертає дискримінант.
* **Функція для знаходження коренів**: використовує значення дискримінанта для обчислення коренів.

**Основна програма**:

* Запитує у користувача значення коефіцієнтів a, b, c.
* Викликає функцію для знаходження коренів і виводить результати.

Текст програми:

import math

def find\_discriminant(a, b, c):

    # Обчислюємо дискримінант

    discriminant = b\*\*2 - 4\*a\*c

    return discriminant

def find\_roots(a, b, c):

    # Знаходимо дискримінант

    D = find\_discriminant(a, b, c)

    if D > 0:

        # Два різних дійсних корені

        x1 = (-b + math.sqrt(D)) / (2\*a)

        x2 = (-b - math.sqrt(D)) / (2\*a)

        return x1, x2

    elif D == 0:

        # Один дійсний корінь

        x = -b / (2\*a)

        return x,

    else:

        # Комплексні корені

        real\_part = -b / (2\*a)

        imaginary\_part = math.sqrt(-D) / (2\*a)

        x1 = complex(real\_part, imaginary\_part)

        x2 = complex(real\_part, -imaginary\_part)

        return x1, x2

# Приклад використання

a = float(input("Введіть коефіцієнт a: "))

b = float(input("Введіть коефіцієнт b: "))

c = float(input("Введіть коефіцієнт c: "))

# Виклик функції та виведення результату

roots = find\_roots(a, b, c)

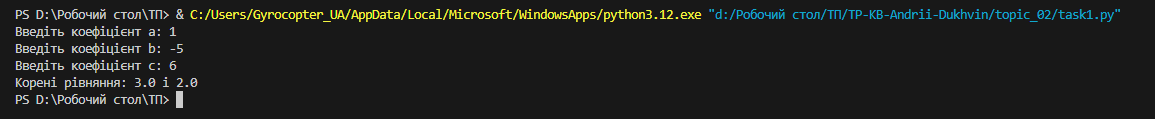
if len(roots) == 1:

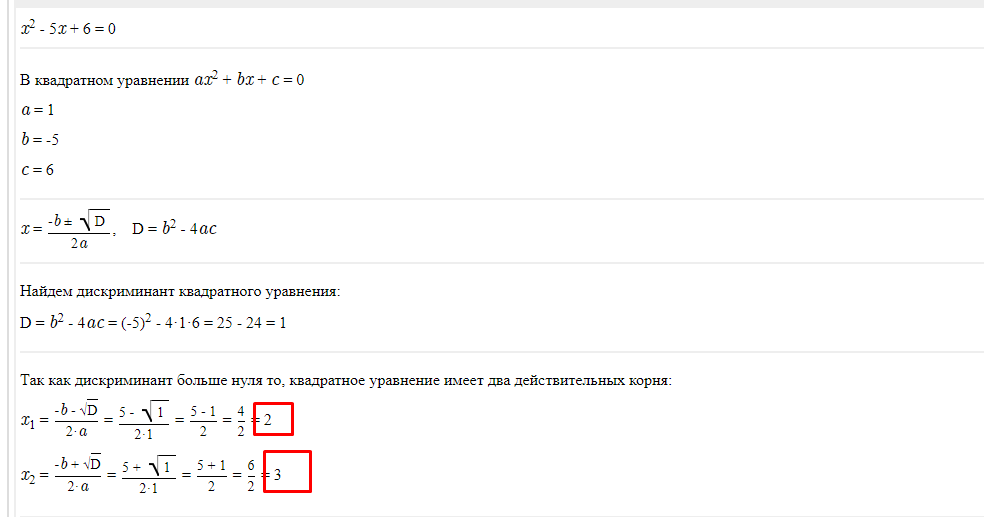
    print(f"Корінь рівняння: {roots[0]}")

else:

    print(f"Корені рівняння: {roots[0]} і {roots[1]}")

Перевірка:





**Завдання 2**

Написати програму калькулятор використовуючи **if else** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід роботи

**Визначення задачі**:

Створити простий калькулятор, який виконує основні арифметичні операції (додавання, віднімання, множення, ділення).

**Структура програми**:

* Реалізувати функції для кожної арифметичної операції: add, subtract, multiply, divide.
* Функція divide повинна обробляти випадок ділення на нуль.

**Написання функцій**:

* **Функція add**: приймає два аргументи і повертає їхню суму.
* **Функція subtract**: приймає два аргументи і повертає результат віднімання другого з першого.
* **Функція multiply**: приймає два аргументи і повертає їхній добуток.
* **Функція divide**: перевіряє, чи другий аргумент не дорівнює нулю, і повертає результат ділення або повідомлення про помилку.

**Основна програма**:

* Запитує у користувача ввести два числа та вибрати операцію.
* Використовує умовні конструкції if-elif-else, щоб викликати відповідну функцію на основі вибраної операції.
* Виводить результат виконаної операції або повідомлення про помилку для невідомої операції.

Текст програми:

# Функції для операцій

def add(a, b):

    return a + b

def subtract(a, b):

    return a - b

def multiply(a, b):

    return a \* b

def divide(a, b):

    if b != 0:

        return a / b

    return "Ділення на нуль!"

# Основна програма

def calculator():

    num1 = float(input("Введіть перше число: "))

    num2 = float(input("Введіть друге число: "))

    operation = input("Оберіть операцію (+, -, \*, /): ")

    if operation == '+':

        print("Результат:", add(num1, num2))

    elif operation == '-':

        print("Результат:", subtract(num1, num2))

    elif operation == '\*':

        print("Результат:", multiply(num1, num2))

    elif operation == '/':

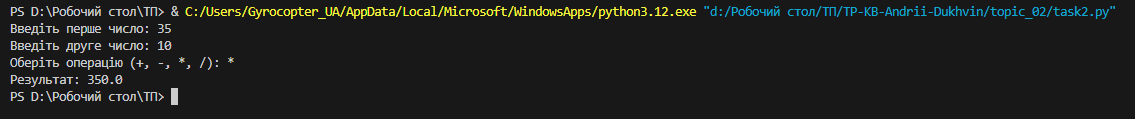
        print("Результат:", divide(num1, num2))

    else:

        print("Невідома операція")

calculator()

Перевірка:





**Завдання 3**

Написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в окремій функції.

Хід роботи

**Визначення задачі**: Розробити калькулятор, який виконує основні арифметичні операції (додавання, віднімання, множення, ділення) за допомогою конструкції match.

**Структура програми**:

* Реалізувати функції для кожної арифметичної операції: add, subtract, multiply, divide.
* У функції divide обробити випадок ділення на нуль, щоб уникнути помилок.

**Написання функцій**:

* **Функція add**: приймає два аргументи і повертає їхню суму.
* **Функція subtract**: приймає два аргументи і повертає результат віднімання другого з першого.
* **Функція multiply**: приймає два аргументи і повертає їхній добуток.
* **Функція divide**: перевіряє, чи другий аргумент не дорівнює нулю, і повертає результат ділення або повідомлення про помилку.

**Основна програма**:

* Запитує у користувача ввести два числа та вибрати арифметичну операцію.
* Використовує конструкцію match для вибору відповідної функції на основі вибраної операції.
* Виводить результат виконаної операції або повідомлення про помилку для невідомої операції.

Текст програми:

# Функції для операцій

def add(a, b):

    return a + b

def subtract(a, b):

    return a - b

def multiply(a, b):

    return a \* b

def divide(a, b):

    if b != 0:

        return a / b

    return "Помилка: ділення на нуль!"

# Основна програма

def calculator():

    num1 = float(input("Введіть перше число: "))

    num2 = float(input("Введіть друге число: "))

    operation = input("Оберіть операцію (+, -, \*, /): ")

    match operation:

        case '+':

            print("Результат:", add(num1, num2))

        case '-':

            print("Результат:", subtract(num1, num2))

        case '\*':

            print("Результат:", multiply(num1, num2))

        case '/':

            print("Результат:", divide(num1, num2))

        case \_:

            print("Невідома операція")

calculator()

Перевірка:  
