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

[Звіт до Теми №1 1](#_Toc1714491368)

[Перетворення рядка 2](#_Toc291606450)

[Суть завдання: 2](#_Toc906207742)

[Хід виконання завдання: 2](#_Toc28400690)

[Код завдання: 2](#_Toc835401789)

[GitHub: 2](#_Toc397682801)

[Перевірка методів роботи зі строками 3](#_Toc1333418786)

[Суть завдання: 3](#_Toc994143228)

[Хід виконання завдання: 3](#_Toc1253167230)

[Код завдання: 3](#_Toc1132200487)

[GitHub: 4](#_Toc414749543)

[Функція знаходження дискримінанту 4](#_Toc1710666625)

[Суть завдання: 4](#_Toc1153784517)

[Хід виконання завдання: 5](#_Toc292316467)

[Код завдання: 5](#_Toc546931080)

[GitHub: 5](#_Toc2121086711)

[Звіт до Теми №2 6](#_Toc284282487)

[Корені квадратного рівняння 7](#_Toc2068632272)

[Суть завдання: 7](#_Toc184151814)

[Хід виконання завдання: 7](#_Toc2000988243)

[Код завдання: 8](#_Toc1808047718)

[GitHub: 8](#_Toc1081969184)

[Калькулятор if else 9](#_Toc1227727485)

[Суть завдання: 9](#_Toc633289105)

[Хід виконання завдання: 9](#_Toc3423820)

[Код завдання: 10](#_Toc1242961875)

[GitHub: 10](#_Toc163731687)

[Калькулятор match 11](#_Toc969991339)

[Суть завдання: 11](#_Toc2027751563)

[Хід виконання завдання: 11](#_Toc409423591)

[Код завдання: 12](#_Toc1558351811)

[GitHub: 12](#_Toc731373737)

# **Звіт до Теми №1**

**Функції та змінні**

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

## Перетворення рядка

### **Суть завдання:**

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

### **Хід виконання завдання:**

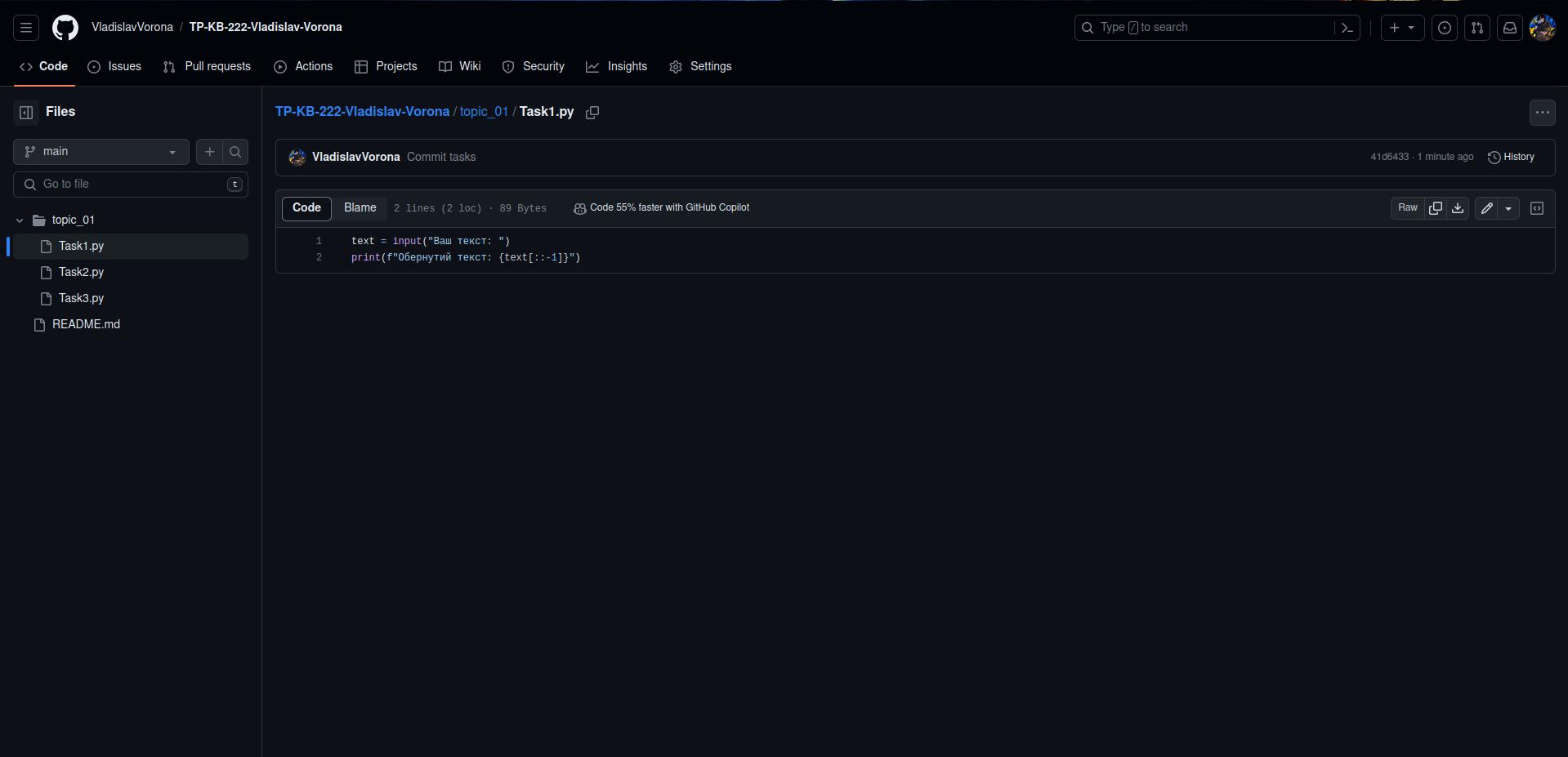
Для виконання цього завдання я використав input() для отримання тексту від користувача і потім записав цей текст в змінну text, після чого використав print() із форматування тексту для зручного відображення інформації. Для обернення рядка я використав оператор [::-1], який створює зріз рядка та включає всі символи в зворотньому порядку.

### **Код завдання:**



### **GitHub:**

Посилання на GitHub: https://github.com/VladislavVorona/TP-KB-222-Vladislav-Vorona/blob/main/topic\_01/Task1.py  
 Знімок екрану з GitHub:



## Перевірка методів роботи зі строками

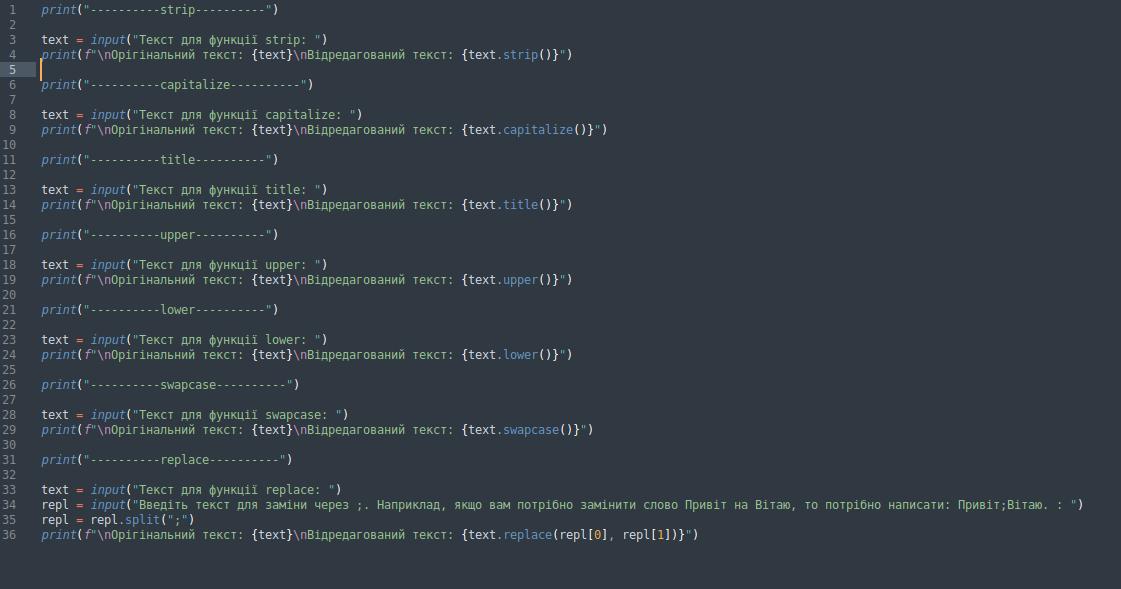
**Суть завдання:**

Необхідно перевірити роботу strip() , capitalize(), title(), upper(), lower(), swapcase(), replace().

### **Хід виконання завдання:**

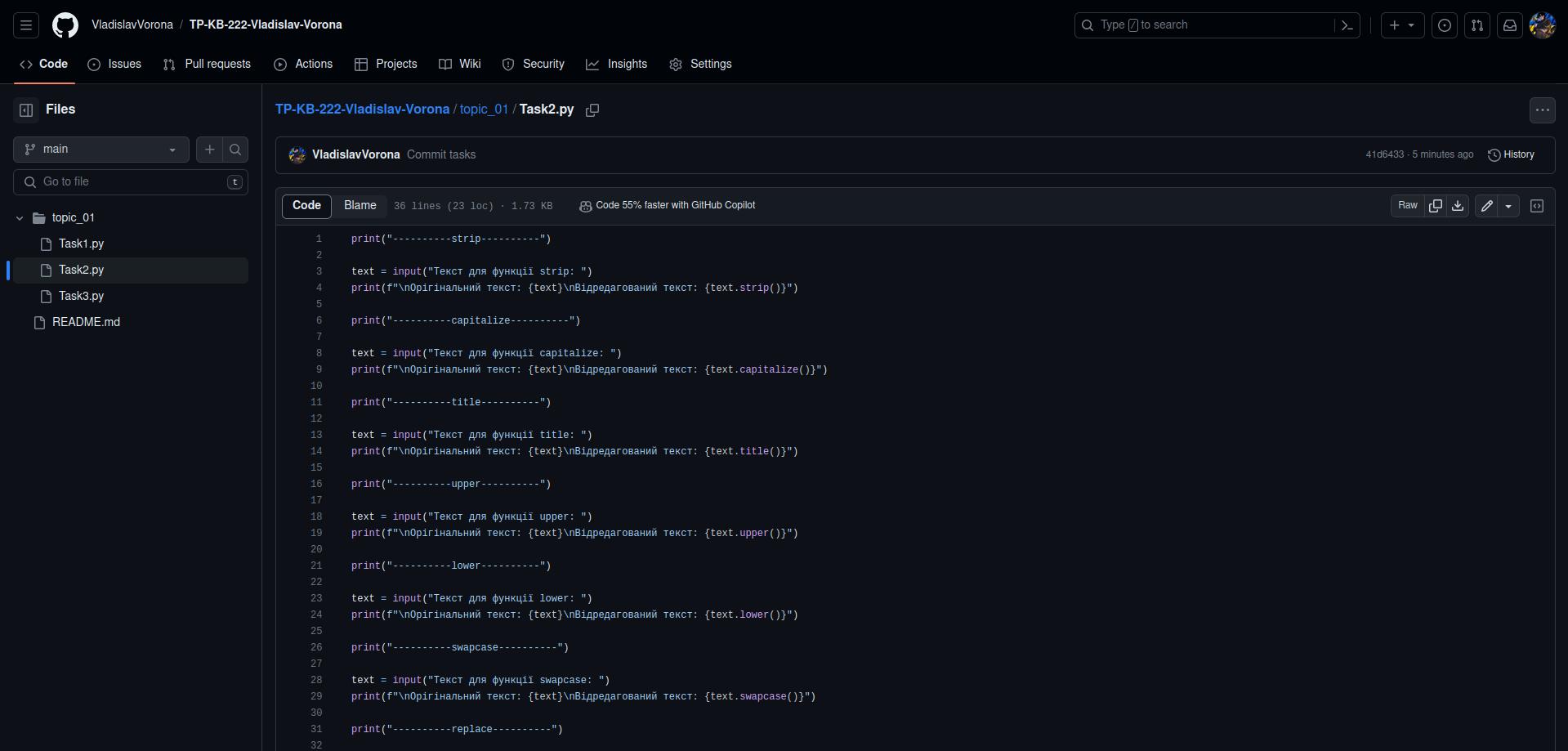
Для виконання цього завдання я використав input() для отримання тексту від користувача і потім записав цей текст в змінну text, після чого використав print() із форматування тексту та перенесенням рядку для зручного відображення інформації. В кожному print() використовується певний метод роботи зі строками, загалом було використано такі методи, як: strip() , capitalize(), title(), upper(), lower(), swapcase(), replace(). Також для демонстрації метода replace(), я використав split() для поділу отриманого тексту від користувача на елементи масиву.

### **Код завдання:**



### **GitHub:**

Посилання на GitHub: https://github.com/VladislavVorona/TP-KB-222-Vladislav-Vorona/blob/main/topic\_01/Task2.py  
 Знімок екрану з GitHub:



## Функція знаходження дискримінанту

**Суть завдання:**

Необхідно написати функцію яка приймає три параметри a,b,c та повертає значення дискримінанту.

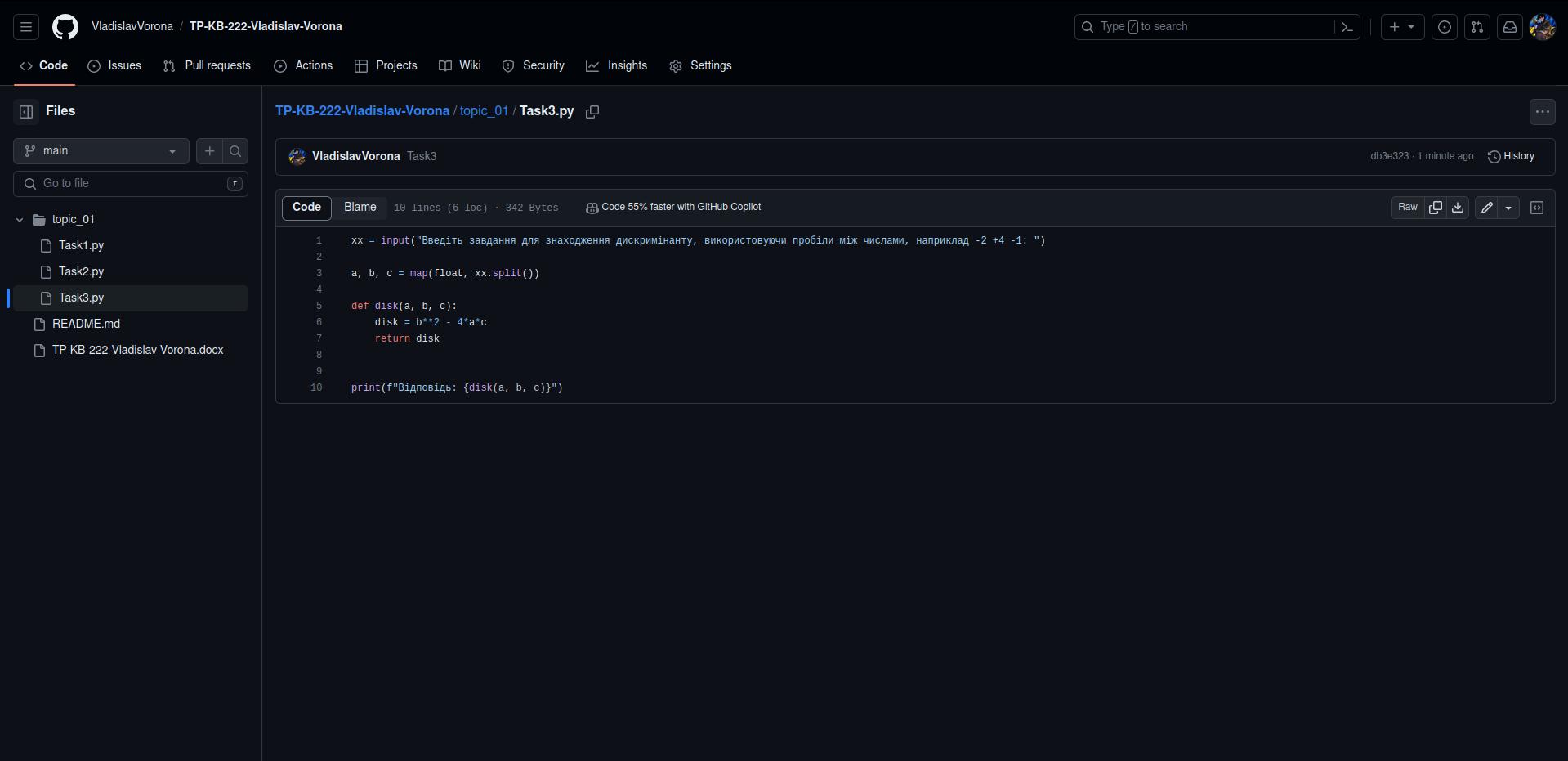
### **Хід виконання завдання:**

Для виконання цього завдання я використав input() для отримання тексту від користувача і потім записав цей текст в змінну xx, після чого я розділив отриманний текст на елементи масиву за допомогою split() та перевів всі значення в float за допомогою функції map() та типізатора float, після чого використав print() із форматуванням тексту для зручного відображення інформації. В середині print(), за допомогою форматування, викликав функцію disk та передав їй значення a, b, c, після чого виконав арифметичні дії, які потрібні для знаходження дискримінанту, використовуючи таку формулу: b^2-4ac, в середині функції, та повернув відповідь за допомогою return.

### **Код завдання:**

### **GitHub:**

Посилання на GitHub: https://github.com/VladislavVorona/TP-KB-222-Vladislav-Vorona/blob/main/topic\_01/Task3.py  
 Знімок екрану з GitHub:



# **Звіт до Теми №2**

**Умовний перехід**

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

## Корені квадратного рівняння

### **Суть завдання:**

Потрібно написати функції пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи

### **Хід виконання завдання:**

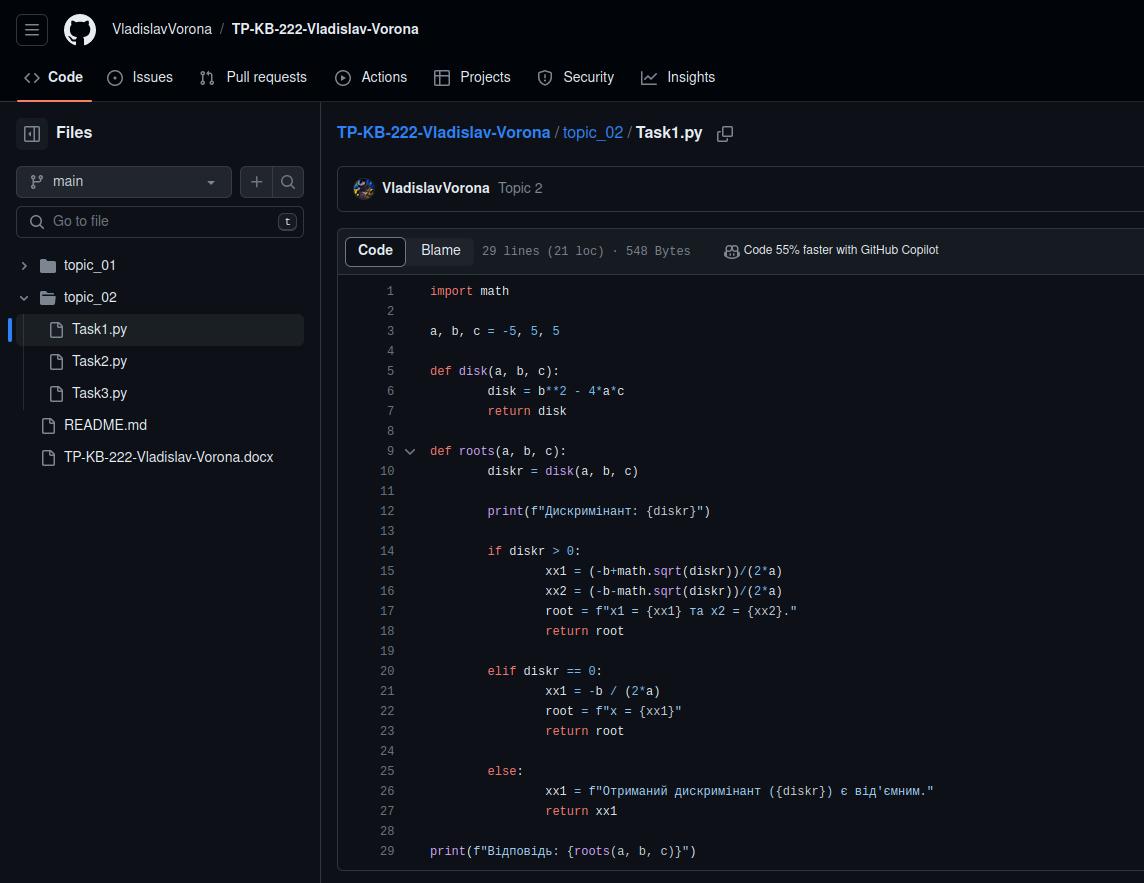
Для виконання цього завдання я імпортував модуль math, який додає можливість використовувати математичні функції, після чого створив змінні та заніс в їх числа. Потім я скопіював функцію розрахунку розрахування дискримінанту із минулої теми і додав її до коду, після чого почав написання функції для пошуку коренів, ця функція отримує змінні, які я вказав раніше, та викликає функцію розрахування дискримінанту та передає їй ці змінні, після чого заносить отриманий результат у змінну diskr, потім виводить за допомогою print() отриманий дискримінант, після чого я використав умову “if diskr > 0:”, якщо ця умова справджується, то починається розрахування коренів за формулою x = (-b±√D)/(2\*a), для розрахунків використовується функція модуля math - sqrt, яка розраховує корінь від числа. Після проведення розрахунків корінь заноситься в змінну і починається розрахунок другого корня, після чого я створив змінну із форматуванням тексту для гарного відображення інформації, після чого повернув цю змінну і за допомогою print() і форматування тексту, вивів корені. Якщо виконується умова “elif diskr == 0:”, то корінь вираховується за формулою -b/(2\*a) та заноситься в змінну, після чого я створив змінну із форматуванням тексту і повернув цю змінну в print() із форматуванням. Якщо жодна із попередніх умов не виконується, то виконується код, який повертає текст, де вказано, що отриманий дискримінант є від’ємним і не може бути розрахованим, після чого відобразив це за допомогою print() і форматування.

### **Код завдання:**



### **GitHub:**

Посилання на GitHub: https://github.com/VladislavVorona/TP-KB-222-Vladislav-Vorona/blob/main/topic\_02/Task1.py  
 Знімок екрану з GitHub:



## Калькулятор if else

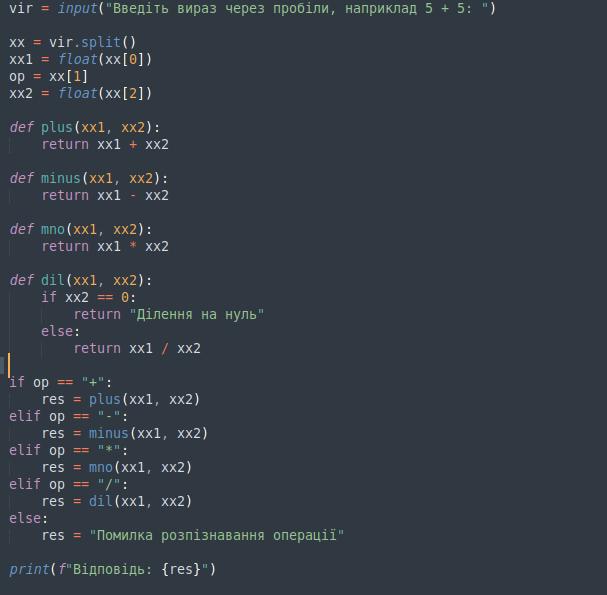
**Суть завдання:**

Потрібно написати програму калькулятор використовуючи **if else** конструкцію.

### **Хід виконання завдання:**

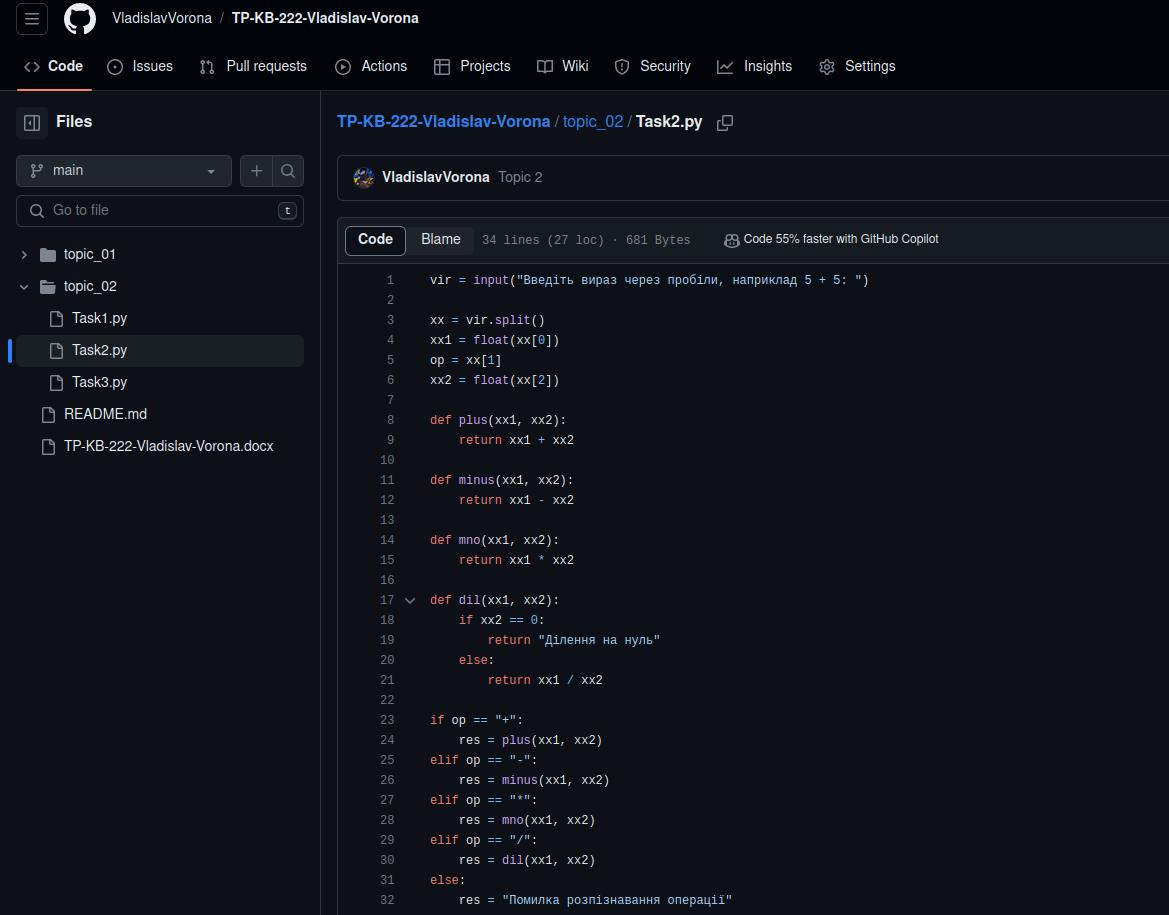
Для виконання цього завдання я спочатку створив змінну та використав input() для отримання завдання від користувача, після чого я отриманий текст перетворив в список значень за допомогою split() та заніс цей список в змінну, після чого почергово створив змінні, дві змінні для значень та одну змінну для операції, після чого заніс в ці змінні потрібну інформацію із списку і форматував текст в float, після чого виконується перевірка на тип операції, якщо змінна із операцією дорівнює "+", то виконується функція із додаванням змінних і після чого результат повертається та зберігається в змінну, теж саме відбувається для інших можливих операцій, таких як віднімання, множення та ділення. Також, якщо виконується операція ділення, то перевіряється чи дорівняє дільник нулю, якщо дільник дорівнює нулю, то в змінну записується текст, що відбулося ділення на ноль, якщо ні, то виконується ділення. Якщо було неправильно вказана операція, то в змінну із результатом записується інформація, що відбулася помилка розпізнавання операції. В кінці я використав print() із форматуванням для відображення результату.

### **Код завдання:**



### **GitHub:**

Посилання на GitHub: https://github.com/VladislavVorona/TP-KB-222-Vladislav-Vorona/blob/main/topic\_02/Task2.py  
 Знімок екрану з GitHub:



## Калькулятор match

**Суть завдання:**

Потрібно написати програму калькулятор використовуючи **match** конструкцію.

### **Хід виконання завдання:**

Для виконання цього завдання я спочатку створив змінну та використав input() для отримання завдання від користувача, після чого я отриманий текст перетворив в список значень за допомогою split() та заніс цей список в змінну, після чого почергово створив змінні, дві змінні для значень та одну змінну для операції, після чого заніс в ці змінні потрібну інформацію із списку і форматував текст в float. Далі я використав конструкцію match, яка дозволяє перевіряти значення змінної та виконувати відповідну операцію в залежності від значення змінної, наприклад, якщо змінна дорівнює "+", то виконується функція із додаванням змінних і після чого результат повертається та зберігається в змінну, теж саме відбувається для інших можливих операцій, таких як віднімання, множення та ділення. Також, якщо виконується операція ділення, то перевіряється чи дорівняє дільник нулю, якщо дільник дорівнює нулю, то в змінну записується текст, що відбулося ділення на ноль, якщо змінна не є нулем, то виконується ділення. Якщо було була вказана невідома операція, то в змінну із результатом записується інформація, що відбулася помилка розпізнавання операції. В кінці я використав print() із форматуванням для відображення результату.

### **Код завдання:**

### **GitHub:**

Посилання на GitHub: https://github.com/VladislavVorona/TP-KB-222-Vladislav-Vorona/blob/main/topic\_02/Task3.py  
 Знімок екрану з GitHub:

