**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:

За допомогою вбудованої функції input(), запросив у користувача рядок через консоль. У функцію передаю один аргумент, а саме рядок з повідомленням для користувача. Результат введення зберігається у глобальну змінну string. Використовуючи слайсинг я перетворюю рядок і одразу передаю його у якості аргумента у функцію print() для виводу на екран.

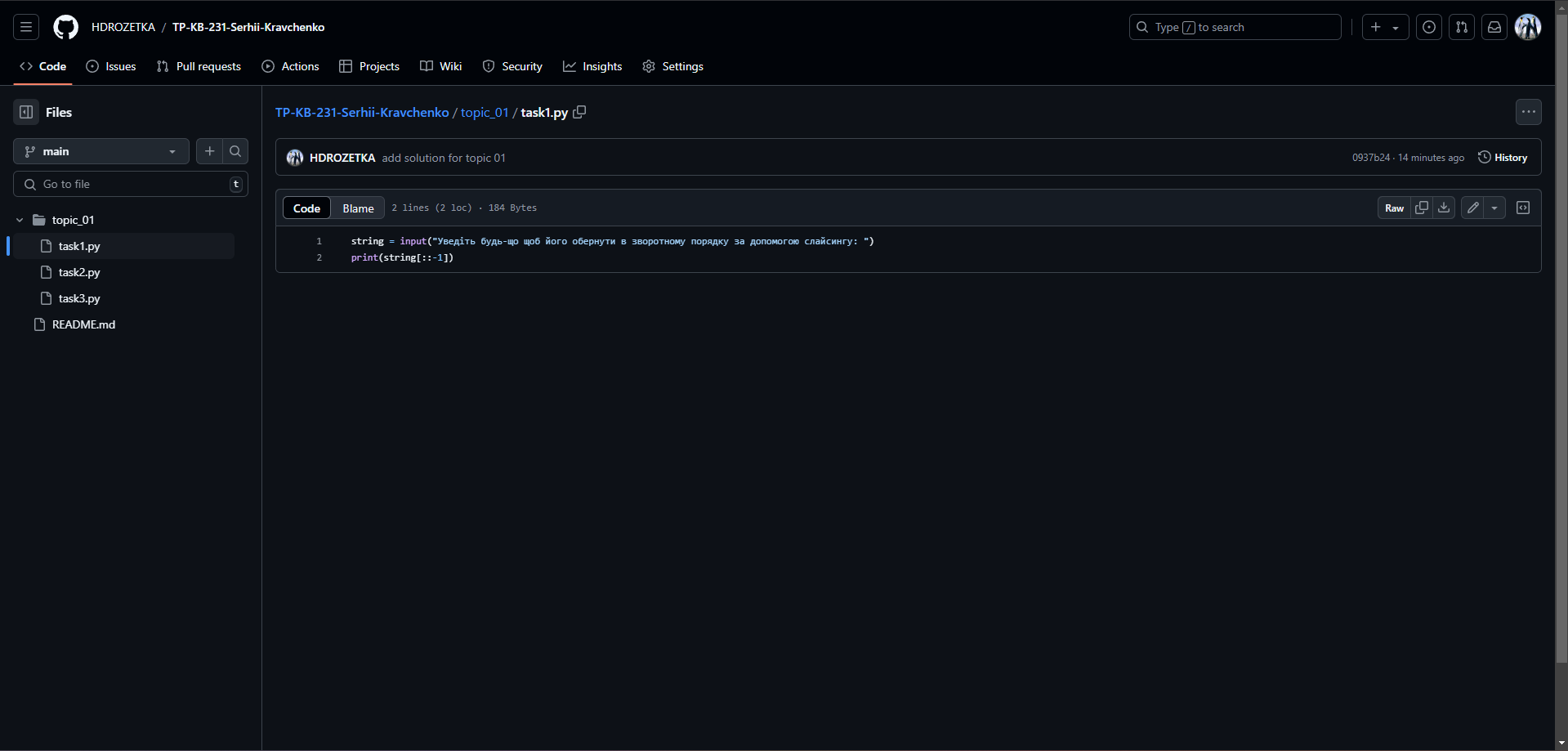
Текст програми:

|  |
| --- |
| string = input("Уведіть будь-що щоб його обернути в зворотному порядку за допомогою слайсингу: ") print(string[::-1]) |

Посилання на github:

<https://github.com/HDROZETKA/TP-KB-231-Serhii-Kravchenko/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github:



**Базові функції для рядків**

Необхідно протестувати роботу вбудованих функцій для рядків strip(), capitalize(), title(), upper(), lower().

Хід виконання:

Використовуючи офіційну документацію Python я дізнався яку саме роботу виконують ці функції. А саме:

.strip() – обрізає рядок по боках. Приймає один необов’язковий аргумент, де вказуються символи що треба видалити.

.capitalize() – робить першу букву рядка великою.

.title() – робить першу букву кожного слова у рядку великою.

.upper() – робить усі букви великими у рядку.

.lower() – робить усі букви маленькими у рядку.

Для кожної функції написав введення користувачем рядка та збереження його у змінній. Для функції .strip() передається аргумент «1234567890», що означає що мають бути видалені усі цифри на початку та у кінці. Зробив вивід модифікованих рядків на екран.

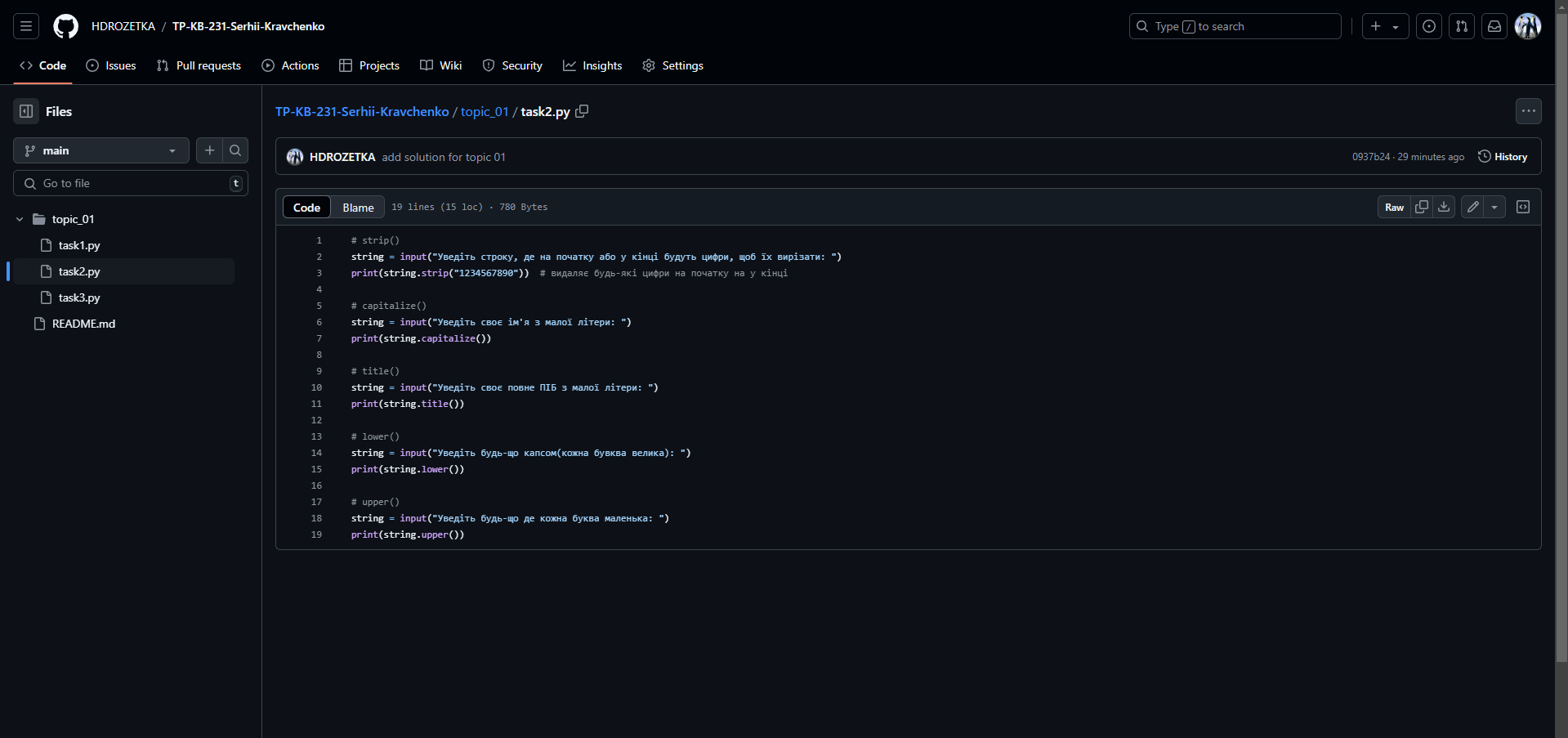
Текст програми:

|  |
| --- |
| # strip() string = input("Уведіть строку, де на початку або у кінці будуть цифри, щоб їх вирізати: ") print(string.strip("1234567890")) # видаляє будь-які цифри на початку на у кінці  # capitalize() string = input("Уведіть своє ім'я з малої літери: ") print(string.capitalize())  # title() string = input("Уведіть своє повне ПІБ з малої літери: ") print(string.title())  # lower() string = input("Уведіть будь-що капсом(кожна бувква велика): ") print(string.lower())  # upper() string = input("Уведіть будь-що де кожна буква маленька: ") print(string.upper()) |

Посилання на github:

<https://github.com/HDROZETKA/TP-KB-231-Serhii-Kravchenko/blob/main/topic_01/task2.py>

Знімок екрану з посилання на github:

****

**Перетворення рядка**

Необхідно створити функцію котра обчислює Дискримінант.

Хід виконання завдання:

За допомогою вбудованої функції input(), запросив у користувача кожен коефіцієнт квадратного рівняння та зберіг їх у відповідні змінні a, b, c. Вище ініціалізував функцію find\_discriminator. За допомогою операторів \*\*, \*, - виконав обчислення Дискримінанту та повернув значення через return. Функція не потребує передачі аргументів для роботи, оскільки змінні a, b, c є глобальними.

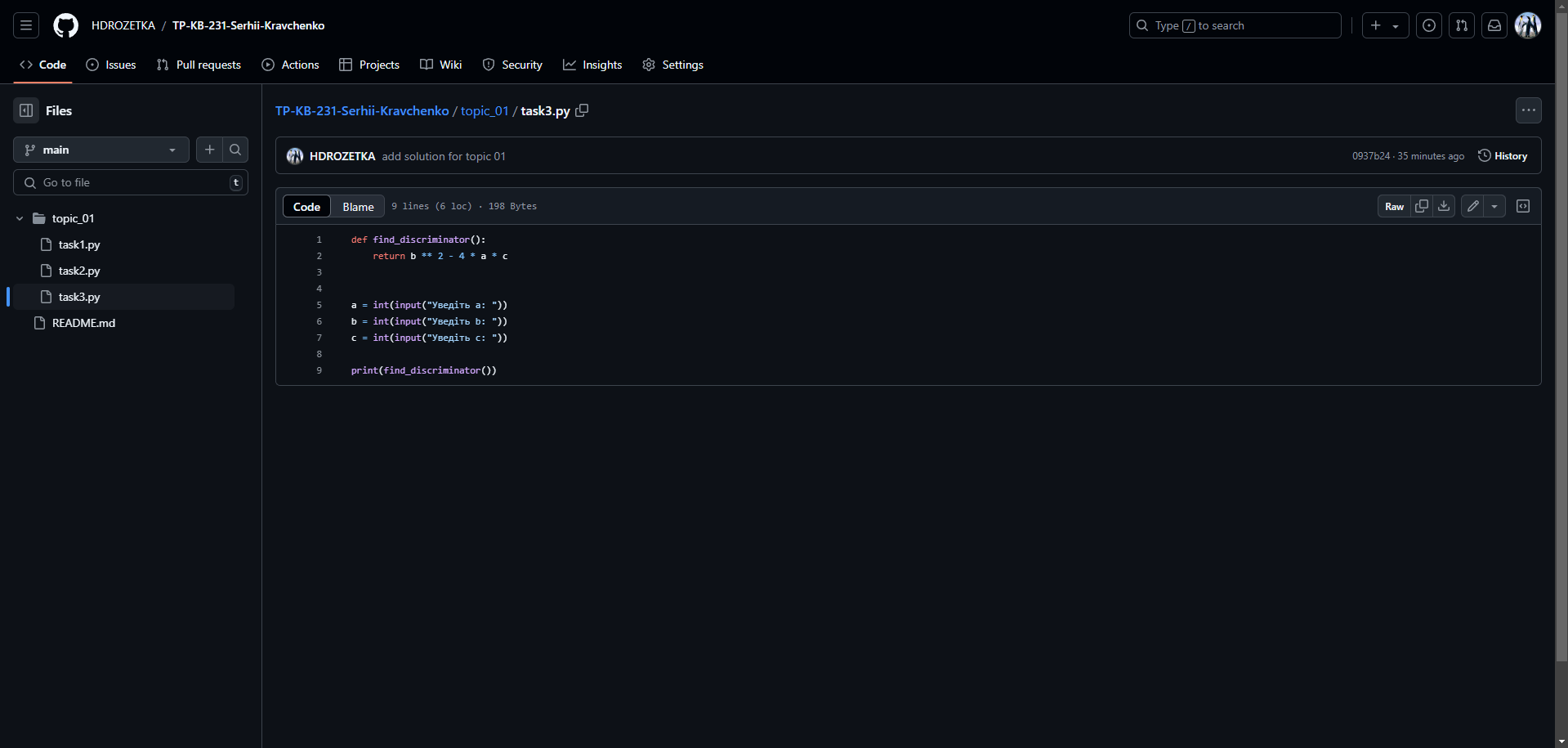
Текст програми:

|  |
| --- |
| def find\_discriminator():  return b \*\* 2 - 4 \* a \* c   a = int(input("Уведіть a: ")) b = int(input("Уведіть b: ")) c = int(input("Уведіть c: "))  print(find\_discriminator()) |

Посилання на github:

<https://github.com/HDROZETKA/TP-KB-231-Serhii-Kravchenko/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github:

****

Звіт до Теми №2

Умовні переходи

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

**Пошук коренів квадратного рівняння враховуючи значення дискримінанту**

Необхідно створити програму котра буде знаходити корені квадратного рівняння. Вона має обробляти усі значення дискримінанту.

Хід виконання завдання:

За допомогою вбудованої функції input(), запросив у користувача значення коефіцієнтів квадратного рівняння через консоль та зберіг їх у відповідні змінні a\_str, b\_str, c\_str. Вище ініціалізував функцію для знаходження коренів. Спочатку вона обчислює дискримінант, якщо він менше нуля(від’ємний) ми повертаємо None, якщо він більше або дорівнює нулю, то виконуємо знаходження коренів x1, x2 та повертаємо їх у кортежі. Потім я зробив перевірку результату. Якщо результат None, ми виводимо напис «немає коренів», якщо їх знайдено, то виводимо кортеж коренів.

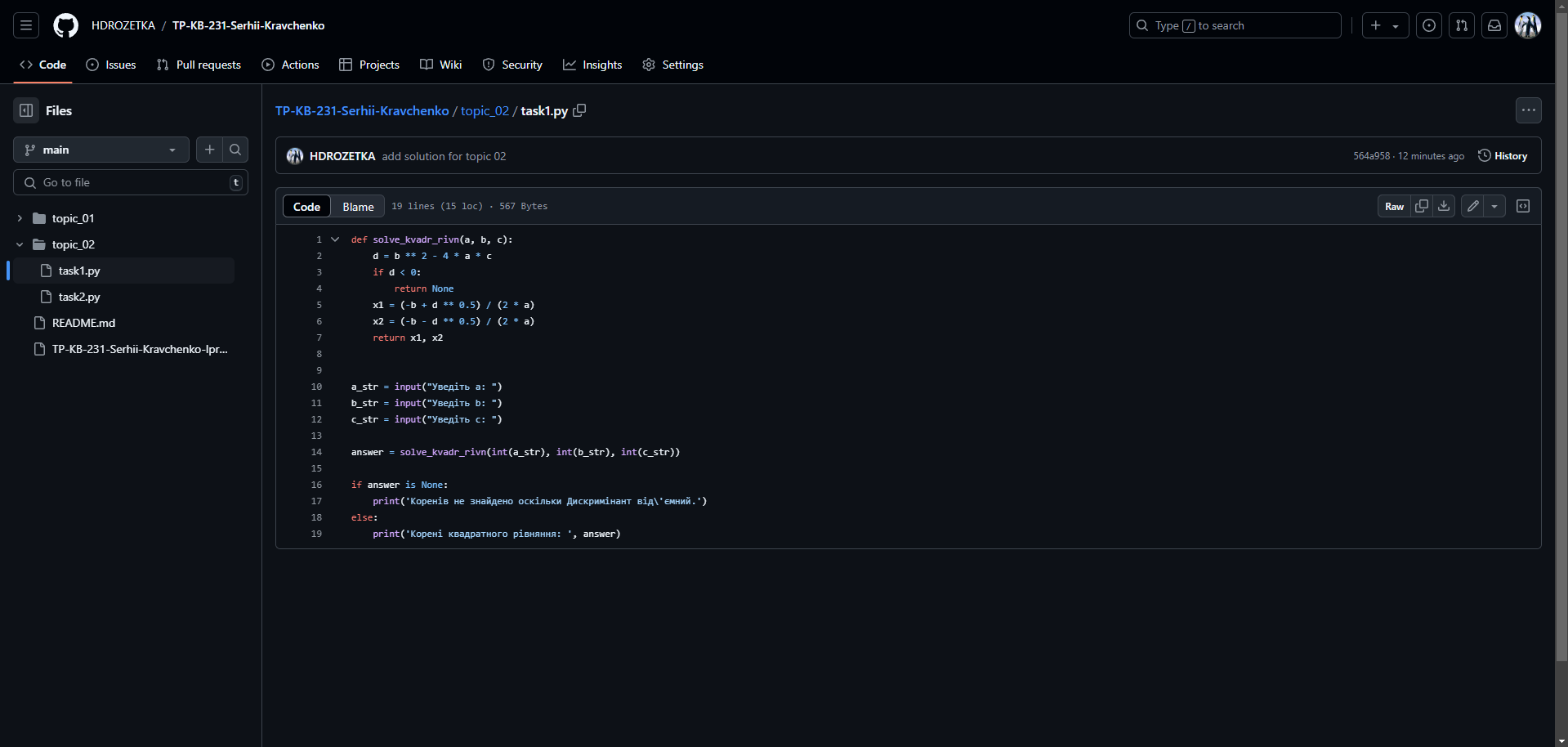
Текст програми:

|  |
| --- |
| def solve\_kvadr\_rivn(a, b, c):  d = b \*\* 2 - 4 \* a \* c  if d < 0:  return None  x1 = (-b + d \*\* 0.5) / (2 \* a)  x2 = (-b - d \*\* 0.5) / (2 \* a)  return x1, x2   a\_str = input("Уведіть a: ") b\_str = input("Уведіть b: ") c\_str = input("Уведіть c: ")  answer = solve\_kvadr\_rivn(int(a\_str), int(b\_str), int(c\_str))  if answer is None:  print('Коренів не знайдено оскільки Дискримінант від\'ємний.') else:  print('Корені квадратного рівняння: ', answer) |

Посилання на github:

<https://github.com/HDROZETKA/TP-KB-231-Serhii-Kravchenko/blob/main/topic_02/task1.py>

Знімок екрану з посилання на github:

****

**Програма калькулятор**

Я ініціалізував дві функції calculator\_if, calculator\_match. Вони приймають аргументи a, b, action. Для функції calculator\_if я зробив перевірку дії action за допомогою опереаторів if, elif, а для calculator\_match зробив на основі match

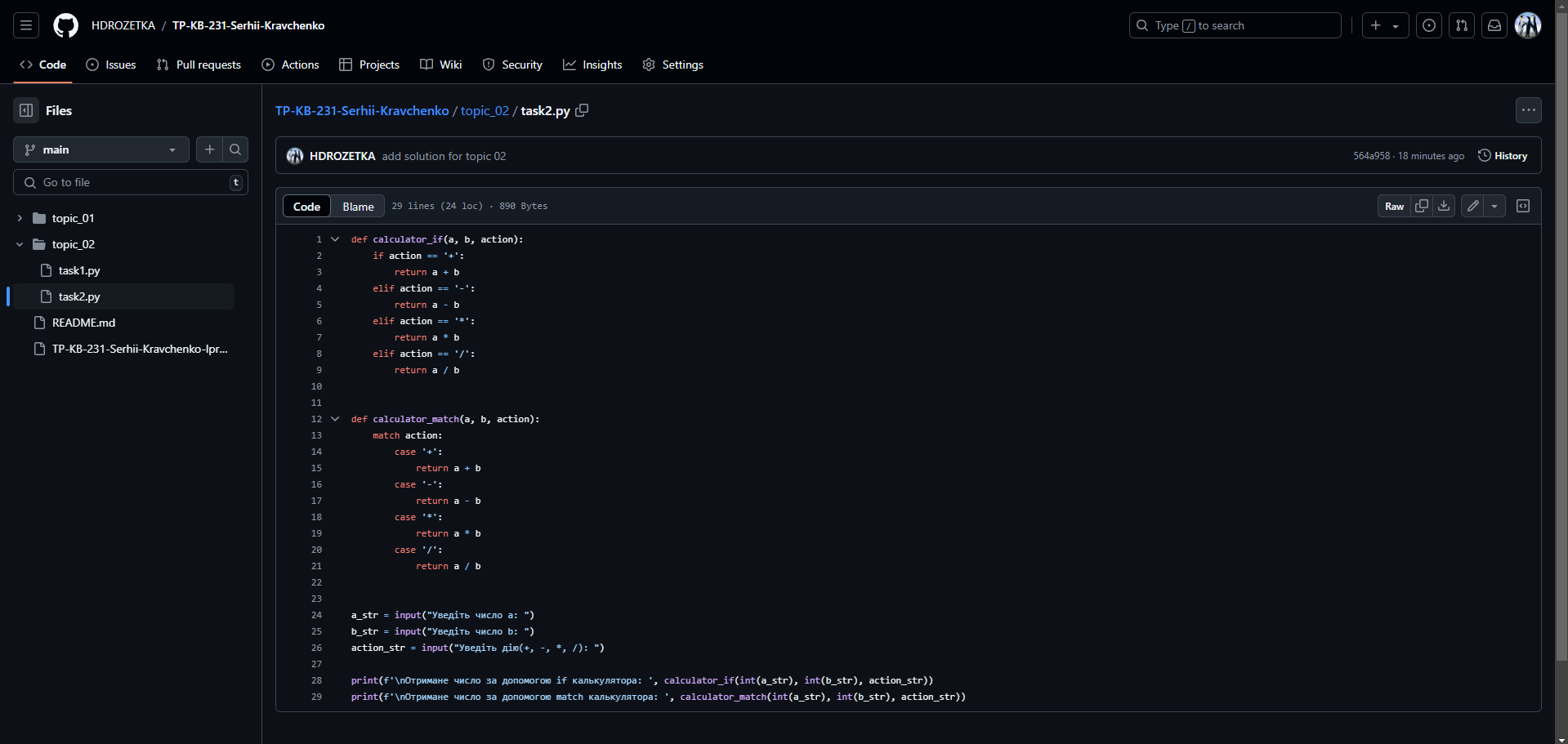
Текст програми:

|  |
| --- |
| def calculator\_if(a, b, action):  if action == '+':  return a + b  elif action == '-':  return a - b  elif action == '\*':  return a \* b  elif action == '/':  return a / b   def calculator\_match(a, b, action):  match action:  case '+':  return a + b  case '-':  return a - b  case '\*':  return a \* b  case '/':  return a / b   a\_str = input("Уведіть число a: ") b\_str = input("Уведіть число b: ") action\_str = input("Уведіть дію(+, -, \*, /): ")  print(f'\nОтримане число за допомогою if калькулятора: ', calculator\_if(int(a\_str), int(b\_str), action\_str)) print(f'\nОтримане число за допомогою match калькулятора: ', calculator\_match(int(a\_str), int(b\_str), action\_str)) |

Посилання на github:

<https://github.com/HDROZETKA/TP-KB-231-Serhii-Kravchenko/blob/main/topic_02/task2.py>

Знімок екрану з посилання на github:

****