**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:**

1. Програма запитує рядок;
2. Інвертування рядка з кроком -1, від кінця до початку;
3. Вивід інвертований рядка в консоль.

**Текст програми:**

|  |
| --- |
| str = input("Enter text for reverse: ")[::-1] |

**Посилання на github:**

https://github.com/Alexxxxo/programmingTechnologies/blob/main/lesson\_02/task1.py

**Знімок екрану з посилання на github:**

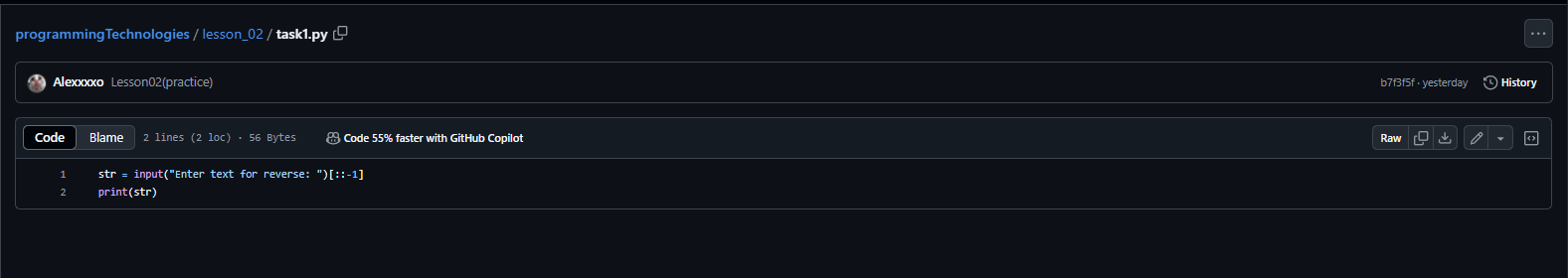


Рисунок 1.1 – Знімок екрану з посилання на GitHub.

**Тестування функцій**

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

**Хід виконання завдання:**

1. Програма запитує рядок;
2. Тестування функції strip() та вивід тестування в консоль;
3. Тестування функції capitalize() та вивід тестування в консоль;
4. Тестування функції title() та вивід тестування в консоль;
5. Тестування функції upper() та вивід тестування в консоль;
6. Тестування функції lower() та вивід тестування в консоль.

**Текст програми:**

|  |
| --- |
| str = input("Enter str: ")  print(str.strip("&")) # removes & characters if they appear at the front of the string and at the end  print(str.capitalize()) # returns a copy of the string and the first letter of the word in uppercase  print(str.title()) # returns a copy of the string, the first letters of the strings will be lowercase, the rest are not  print(str.upper()) # returns upper case strings  print(str.lower()) # returns lower case strings |

**Посилання на github:**

<https://github.com/Alexxxxo/programmingTechnologies/blob/main/lesson_02/task2.py>

**Знімок екрану з посилання на github:**

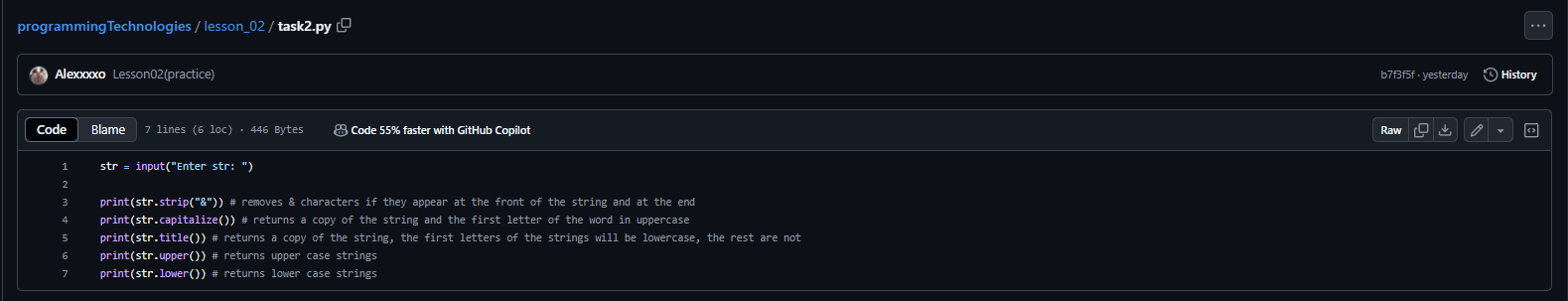


Рисунок 1.2 - Знімок екрану з посилання на GitHub.

**Функція пошуку дискримінанту квадратного рівняння**

Написати функцію пошуку дискримінанту квадратного рівняння.

**Хід виконання завдання:**

1. Імпорт функцій з модуля math;
2. Створення функції для обчислення дискримінанта;
3. Створення функції для розв’язку квадратного рівняння;
4. Якщо дискримінант більше 0, то обчислення два корені, та повертання коренів;
5. Якщо дискримінант дорівнює 0, то обчислення одного кореня, та повертання кореня;
6. Якщо дискримінант менше 0, то повертаємо що коренів немає.

**Текст програми:**

|  |
| --- |
| from math import \*  def discriminant(a, b, c): # Пошук дискримінанта      return b\*\*2 - 4 \* a \* c  def quadratic(a, b, c):      D = discriminant(a, b, c)        if D > 0: # Якщо дискримінант більше 0, то має два кореня          x1 = (-b + sqrt(D)) / (2\*a)          x2 = (-b - sqrt(D)) / (2\*a)          return x1, x2      elif D == 0: # Якщо дискримінант == 0, то має один корень          x = -b / (2 \* a)          return x, x      else: # Якщо дискримінант < 0, то немає коренів          return "Коренів немає"  print(quadratic(-2, 5, 6)) |

**Посилання на github:**

<https://github.com/Alexxxxo/programmingTechnologies/blob/main/lesson_02/task3.py>

**Знімок екрану з посилання на github:**

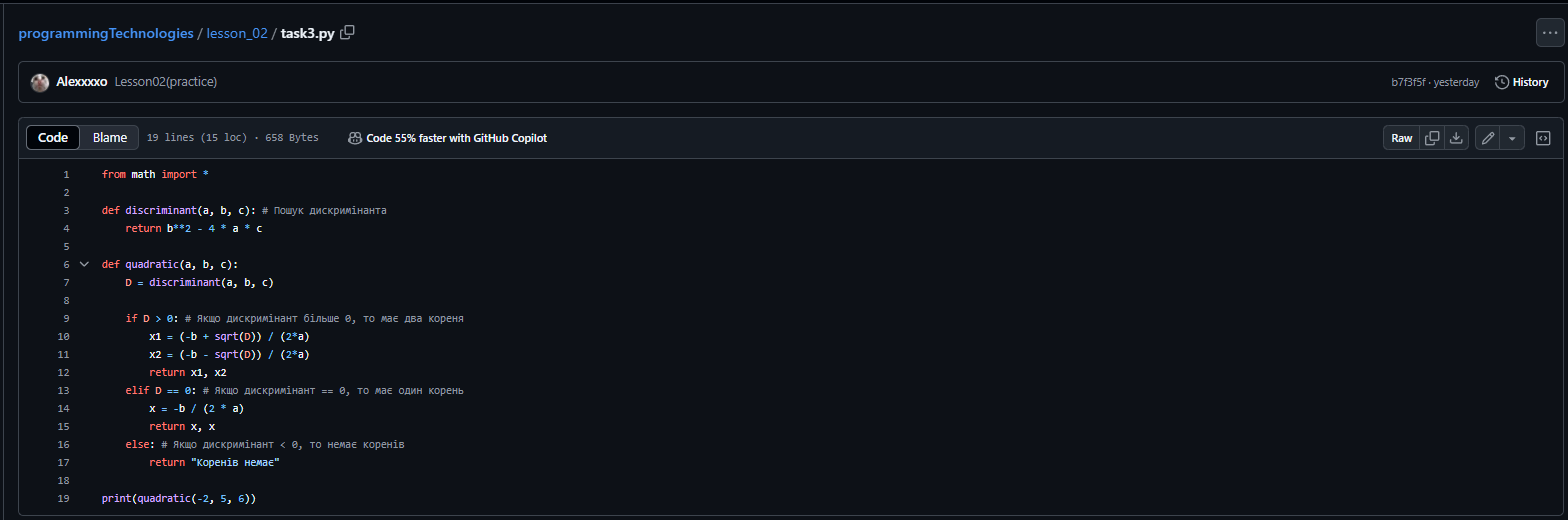


Рисунок 1.3 - Знімок екрану з посилання на GitHub.