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

[Звіт до Теми №1 1](#_Toc679688436)

[Перетворення рядка 2](#_Toc313283489)

[Тестування методів роботи зі строками 3](#_Toc1629618824)

[Функція знаходження дискримінанту 5](#_Toc952032513)

[Звіт до Теми №2 6](#_Toc2099457208)

[Квадратні корені рівняння 7](#_Toc1015750083)

[Калькулятор if else 9](#_Toc1075483826)

[Калькулятор match 11](#_Toc1017553311)

[Звіт до Теми №3 13](#_Toc1479975328)

[Програма калькулятор 13](#_Toc84142368)

[Тестування функцій списків 15](#_Toc306464957)

[Тестування функцій словників 17](#_Toc1331723147)

[Функція пошуку позицій 18](#_Toc1871891604)

# Звіт до Теми №1

Функції та змінні

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

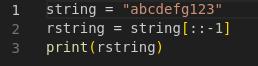
## Перетворення рядка

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

Хід виконання завдання:

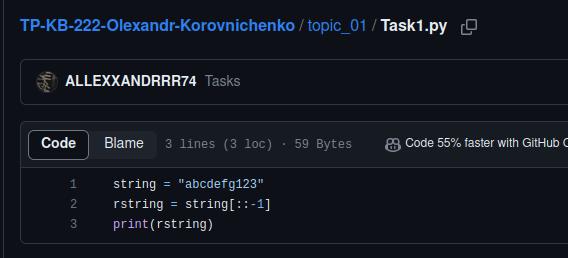
Для виконання цього завдання я використав змінну string. Для обернення рядка я використав оператор [::-1], який створює зріз рядка та включає всі символи в зворотньому порядку та потім вивів результат роботи оператора через print() за допомогою змінної rstring.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_01/Task1.py

Знімок екрану з посилання на github:



## Тестування методів роботи зі строками

Необхідно протестувати роботу strip() , capitalize(), title(), upper(), lower().

Хід виконання завдання :

Для виконання цього завдання я використовував змінні для передавання тексту до потрібних методів роботи зі строками та виводив результат їхного виконання за допомогою print().

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_01/Task2.py

Знімок екрану з посилання на github:



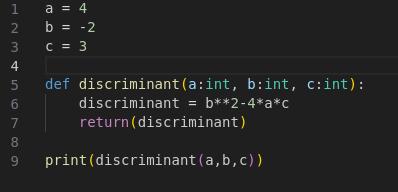
## Функція знаходження дискримінанту

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

Хід виконання завдання :

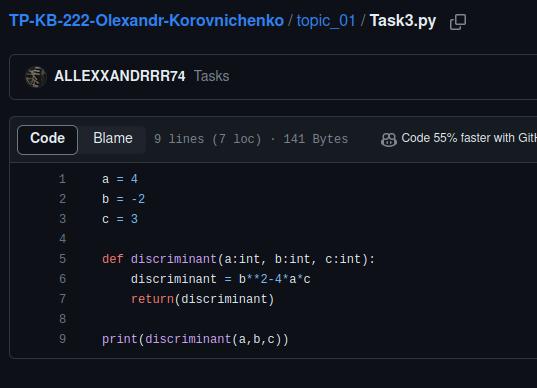
Для виконання цього завдання я використовував змінні для передачі даних до функції discriminant, яку я викликав в print(). Сама функція отримує переданій змінні та за допомогою арифметичних дій обчислює діскрімінант по формулі b^2-4ac, після чого повертає відповідь в print() за допомогою return().

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_01/Task3.py

Знімок екрану з посилання на github:



# Звіт до Теми №2

Функції та змінні

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

## Квадратні корені рівняння

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

Хід виконання завдання:

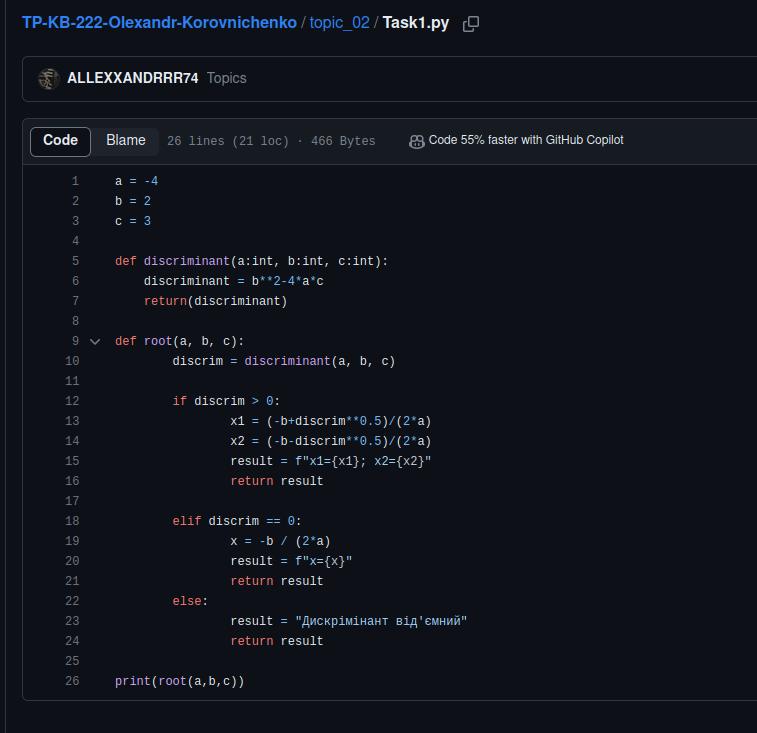
Для виконання цього завдання я скопіював код із минулого завдання та створив функцію для розрахування коренів рівняння. В середині цієї функції я викликаю функцію розрахунку дискримінанту, після чого перевіряю чи більший дискримінант за нуль і якщо так, то розраховую х1 та х2 за формулою і повертаю результат. Якщо дискримінант дорівнює нулю, то я розраховую х за формулою і повертаю відповідь, а якщо дискримінант від’ємний, то виводиться текст, що дискримінант від’ємний.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_02/Task1.py

Знімок екрану з посилання на github:



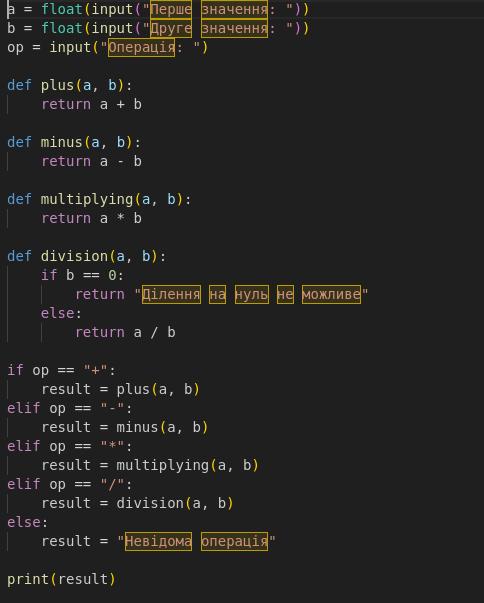
## Калькулятор if else

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

Хід виконання завдання :

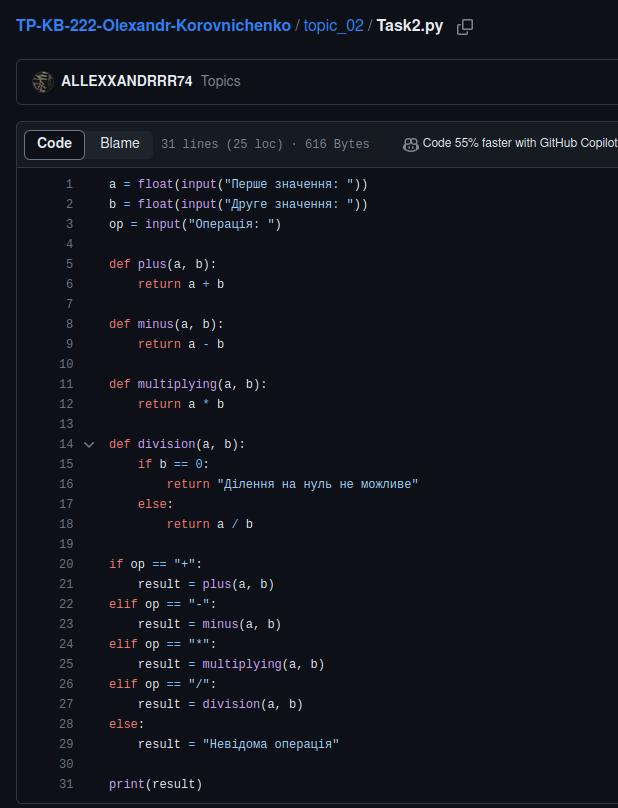
Для виконання цього завдання я створив змінні та заніс в їх інформацію за допомогою input і потім перевіряю операцію за допомогою if elif else і виконую її, викликаючи функцію із цією операцією, а функція повертає результат, який я записую в змінну та виводжу в консоль.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_02/Task2.py

Знімок екрану з посилання на github:



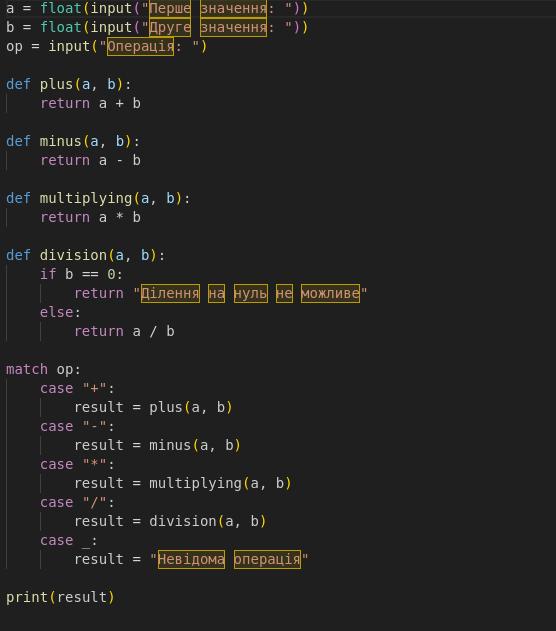
## Калькулятор match

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

Хід виконання завдання :

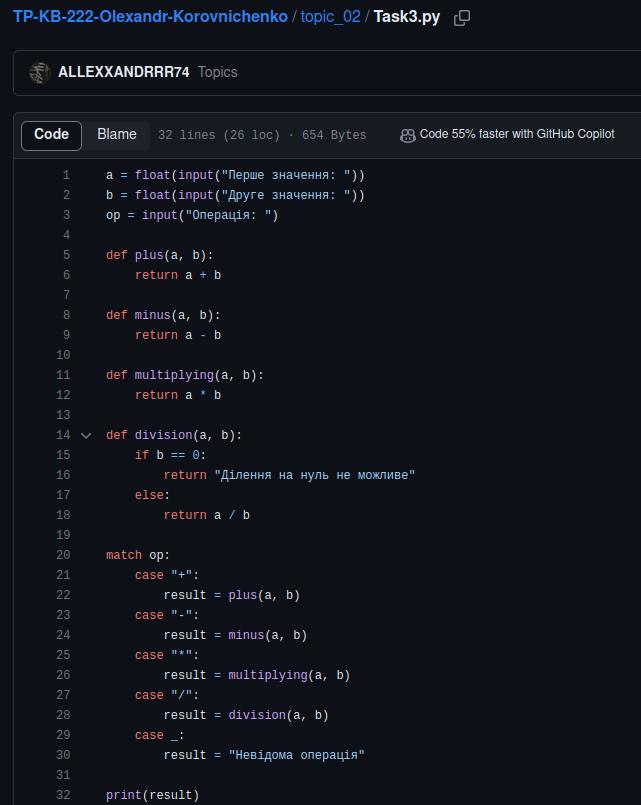
Для виконання цього завдання я створив змінні та заніс в їх інформацію за допомогою input і потім перевіряю операцію за допомогою match і виконую її, викликаючи функцію із цією операцією, а функція повертає результат, який я записую в змінну та виводжу в консоль.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_02/Task3.py

Знімок екрану з посилання на github:



# Звіт до Теми №3

Функції та змінні

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

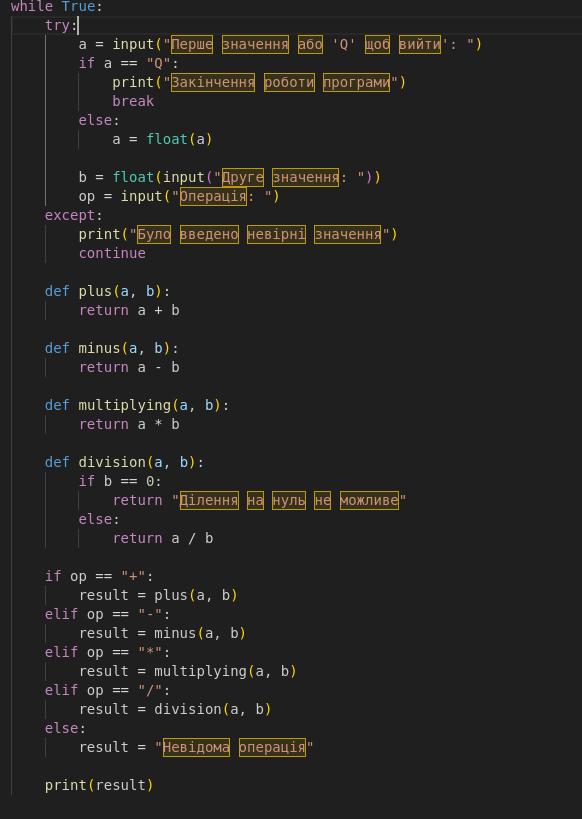
## Програма калькулятор

Написати програму калькулятор з постійними запитами на введення нових даних та операцій. За основу взяти програму калькулятор з попередньої теми. Реалізувати механізм завершення програми після отримання відповідної команди.

Хід виконання завдання:

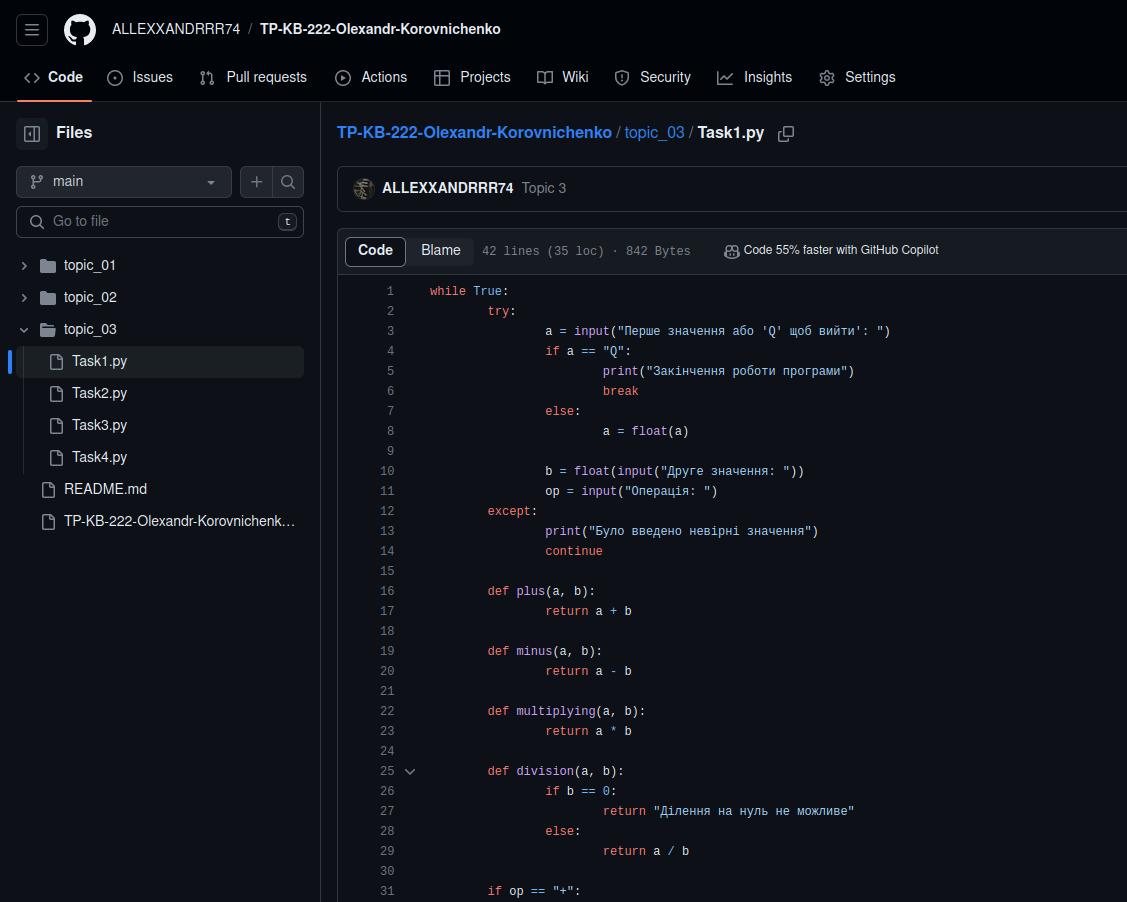
Для виконання цього завдання я скопіював код із минулого завдання та додав вічний цикл до коду і переніс в середину його увесь код. Далі я додав умову, що якщо змінна ‘а’ дорівнює ‘Q’ то закінчувати виконання программи. Також я додав до коду блоки try та except для розпізнавання того, чи правильні вводить данні користувач і якщо при форматуванні змінних в тип float відбувається помилка, то виводиться надпис, що було введено невірні значення і цикл починається спочатку.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_03/Task1.py

Знімок екрану з посилання на github:



## Тестування функцій списків

Написати програму тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy()

Хід виконання завдання :

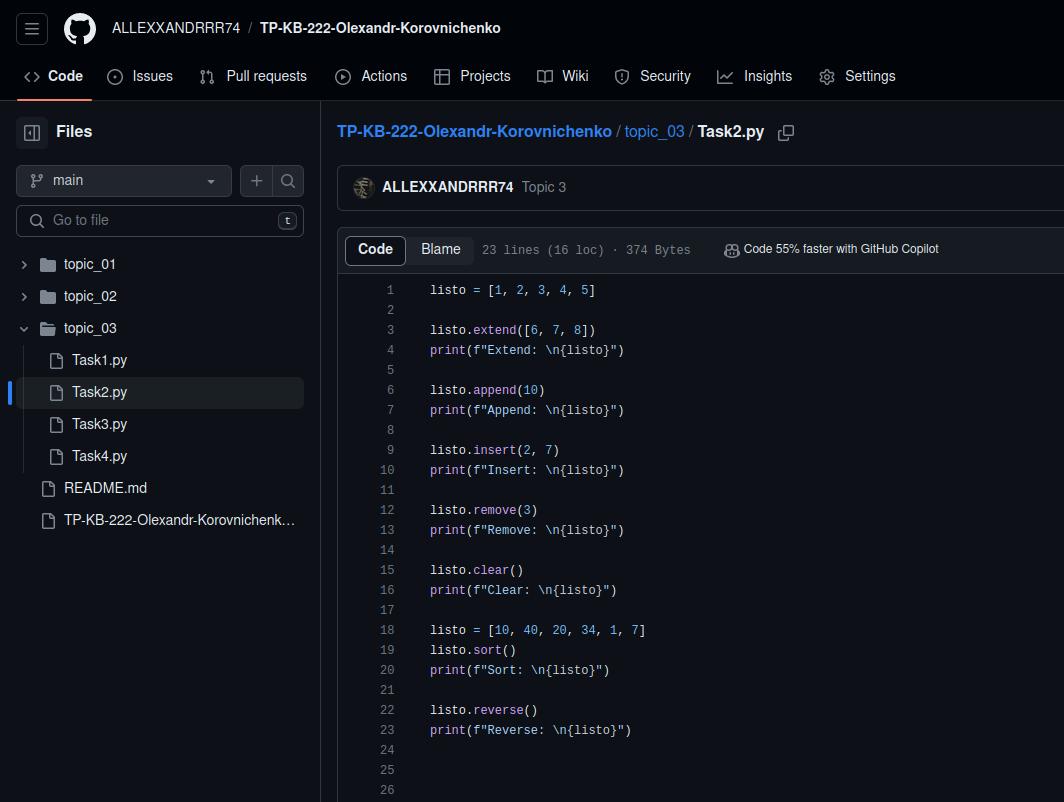
Для виконання цього завдання я створив список listo і потім почергово тестував на йому функції для роботи із списками, серед них: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy() і після виконання кожної функції я виводив оновлений список.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_03/Task2.py

Знімок екрану з посилання на github:



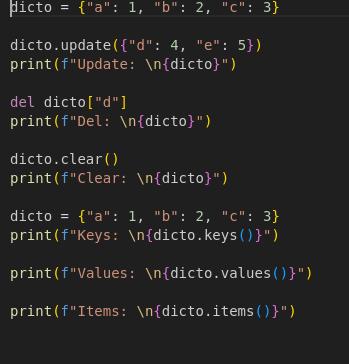
## Тестування функцій словників

Написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items()

Хід виконання завдання :

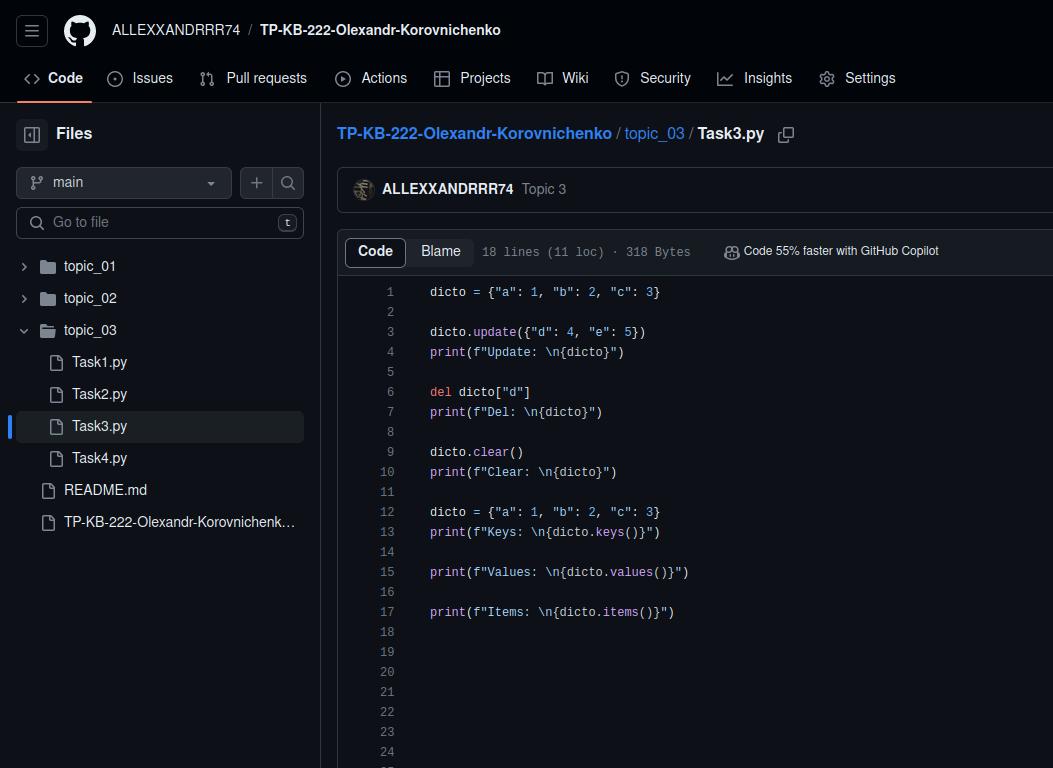
Для виконання цього завдання я створив словник dicto і потім почергово тестував на йому функції для роботи із словниками, серед них: update(), del(), clear(), keys(), values(), items() і після виконання кожної функції я виводив оновлений словник.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_03/Task3.py

Знімок екрану з посилання на github:



## Функція пошуку позицій

Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

Хід виконання завдання :

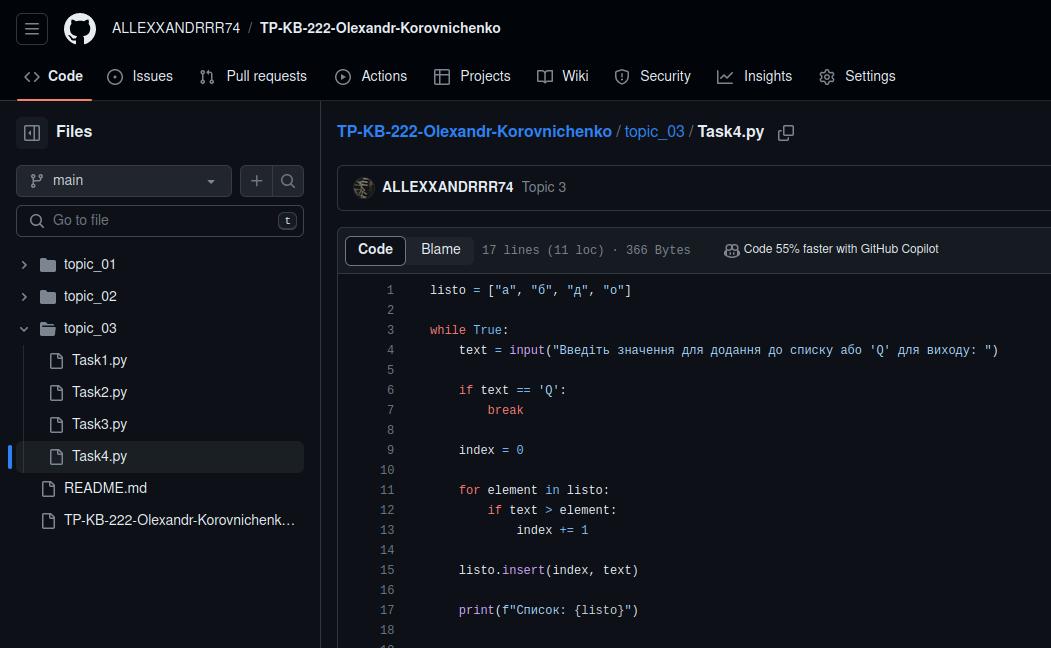
Для виконання цього завдання я створив список і заніс в його відсортовані дані. Далі я створив безкінечний цикл в змінну, в яку користувач вводить значення для додавання в список. Якщо значення дорівнює Q то виконується закінчення роботи циклу і програми. Далі я створив змінну індекс і надав їй значення 0 і після чого створив цикл, який перевіряв всі елементи в списку listo і якщо введене користувачем значення більше за вибраний елемент в списку, то до індексу додається одиниця. Після виконання циклу відбувається занесення в список елементу за потрібним індексом і виведення списку.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_03/Task4.py

Знімок екрану з посилання на github:



# Звіт до Теми №4

Виняткові події

Під час виконання практичного завдання до Теми №4 було надано варіанти рішення до наступних задач:

## Програма калькулятор

Розширити програму калькулятор функцією запитів даних для виконання операцій від користувача, що обробляє виняткові ситуації Розширити функцію ділення обробкою виняткової ситуації ділення на нуль.

Хід виконання завдання:

Для виконання цього завдання я скопіював код калькулятора із минулих завдань і створив дві функції, одна із них отримує числове значення від користувача та якщо користувач вводить невірне значення то за допомогою ValueError код обробляє цю помилку і просить користувача ввести значення знову, а якщо все добре, то повертає відформатоване значення. Друга функція отримує операцію від користувача і перевіряє чи є введена операція в списку, якщо нема, то просить користувача ввести операцію знову, якщо все добре, то повертає операцію. В головному циклі я викликаю ці 2 функції і після виклику функції операції я перевіряю, чи не є вона ‘Q’ і якщо вона є нею, то закінчує виконання програми. Також до функції ділення було додано виняткову подію ZeroDivissionError, яка спрацьовує кожного разу, коли користувач хоче поділити значення на нуль.

Код програми:



**Посилання на github:** https://github.com/ALLEXXANDRRR74/TP-KB-222-Olexandr-Korovnichenko/blob/main/topic\_04/Task1.py

Знімок екрану з github:

