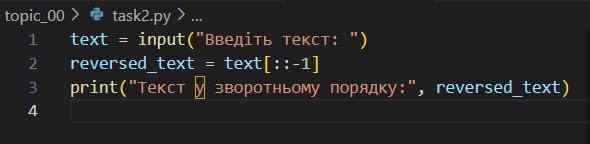
**Звіт про виконання практичних завдань до лекцій з курсу Технології програмування**

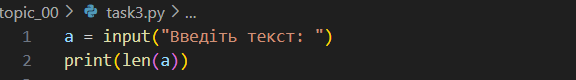
1. **Функції та змінні**

* Повернути рядок в зворотному порядку.

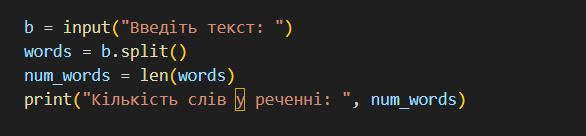


Змінна *text* запитує у користувача текст, який треба повернути в зворотному порядку. Далі створюємо функцію *reversed\_text*, яка буде обертати наш текст у зворотному порядку за допомогою *[:: -1],* що буде обирати елементи в зворотному порядку. За допомогою *print()* виводимо результат.

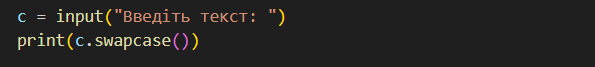
* Протестувати базові функції для рядків.



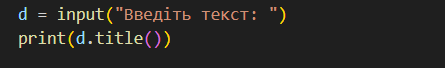
Змінна *а* запитує текст, функція *len()* рахує кількість літер в рядку і за допомогою функції *print()* виводить кількість літер, які ввів користувач.



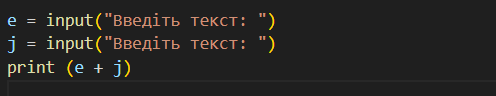
Змінна *b* запитує текст, вводимо змінну words, де функція *split()* буде розбивати введений текст на слова, використовуючи пробіл як спосіб їх розділяти. Змінна *num\_words()* буде викликати words щоб підрахувати їх кількість функцією *len()*.За допомогою *print* виводимо результат.



Змінна *с* запитує у користувача текст, функція *swapcase()* змінює кожну літеру в рядку, великі літери на малі, а малі - на великі, функцією *print()* виводимо результат.

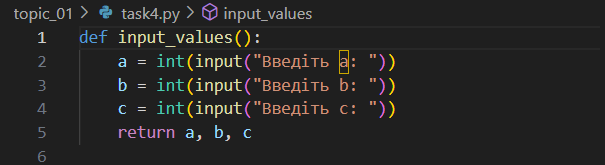


Змінна *d* запитує текст, функція *title()* змінює першу літеру в реченні на велику, функцією *print()* виводимо результат.

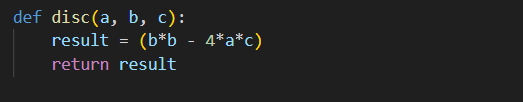


Змінні *e* та *j* запитують текст і в функції та в функції *print()* об’єднуються.

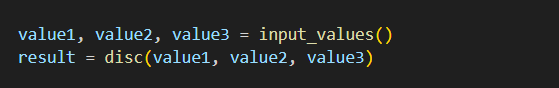
* Написати функцію пошуку Дискримінанту.



У функції *input\_values()* запитуємо числові змінні *a, b, c* та повертаємо значення.



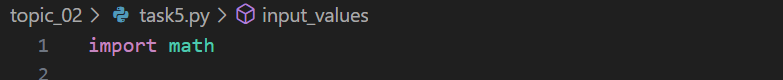
Функція *disc()* працюватиме з введеними значеннями та буде обчислювати дискримінант за формулою та повертати результат.



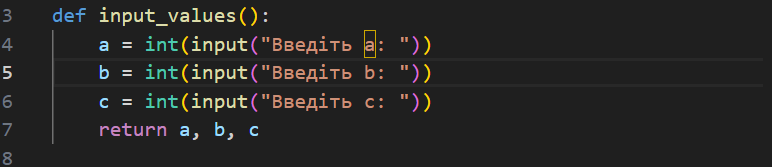
Змінні *value1, value2, value3* отримують значення *a, b, c.* Змінна *result* викликає функцію *disc()* для обчислення дискримінанту з отриманими значеннями *value1, value2, value3*.

1. **Умовні переходи.**

* Виконати пошук коренів квадратного рівняння враховуючи значення дискримінанту.



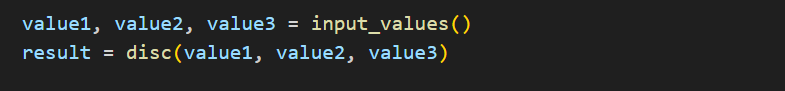
Додамо модуль *math*, він потрібен щоб використовувати знак кореню в обчисленнях.

\

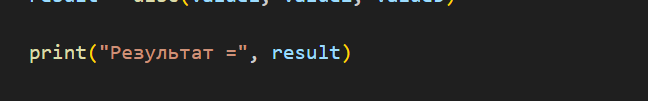
У функції *input\_values()* запитуємо числові змінні *a, b, c* та повертаємо значення.



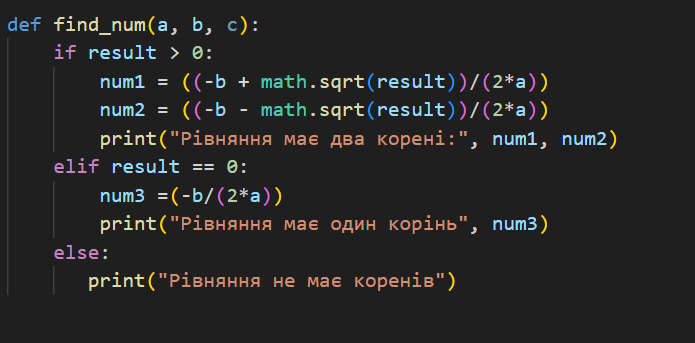
Функція *disc()* працюватиме з введеними значеннями та буде обчислювати дискримінант за формулою та повертати результат.



Змінні *value1, value2, value3* отримують значення a, b, c. Змінна *result* викликає функцію *disc()* для обчислення дискримінанту з отриманими значеннями *value1, value2, value3*.

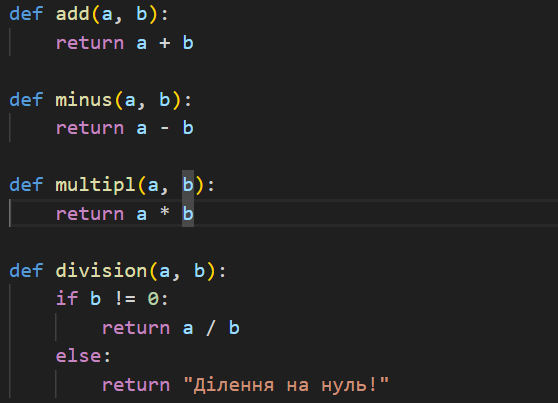


Функція *print()* виводитиме результат.



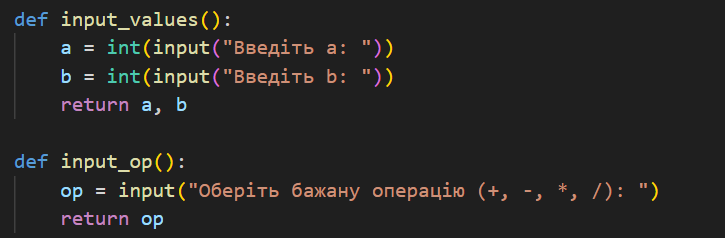
Створимо умову згідно теоремам, якщо результат більше 0, то корені 2 і задамо формулу для їхнього обчислення, якщо результат 0, то корінь 1, запишемо формулу для нього, в усіх інших випадках рівняння не має коренів.

* Програма калькулятор на основі *if elif else*.

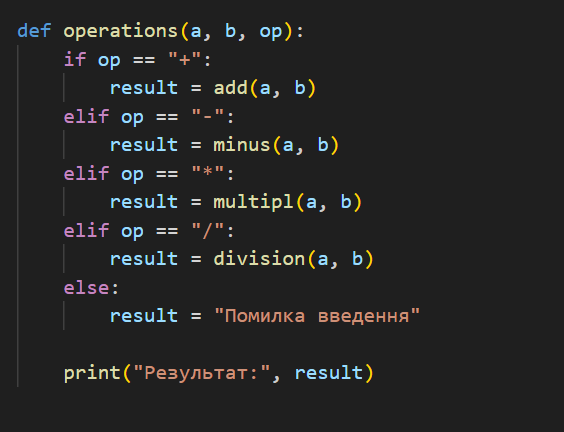


Задамо функціїї, які будуть означати дії з числами, які ми будемо викликати в процесі, додавання *(a + b)*, віднімання *(a – b)*, множення *(a \* b)*, ділення *(a / b)*.

Задамо через *if else*, що якщо *b* не дорівнює нулю, то виконується дія ділення, а якщо дорівнює, то буде виводитись попередження і результату не буде.

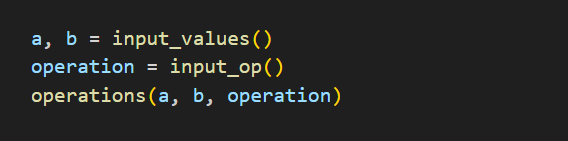


Введемо функцію, яка буде запитувати у користувача числові значення та бажану операцію з ними.



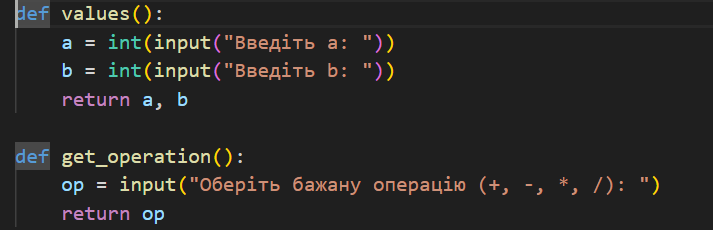
Перейдемо до умови *if elif* та *else*.

Задамо якщо операція +, то викликається функція з додаванням, через elif задамо операцію – з викликом функції віднімання, операцію \* з викликом функції множення та операцію / з викликом функції ділення. В усіх інших випадках буде помилка введення операції. Виведення результату.

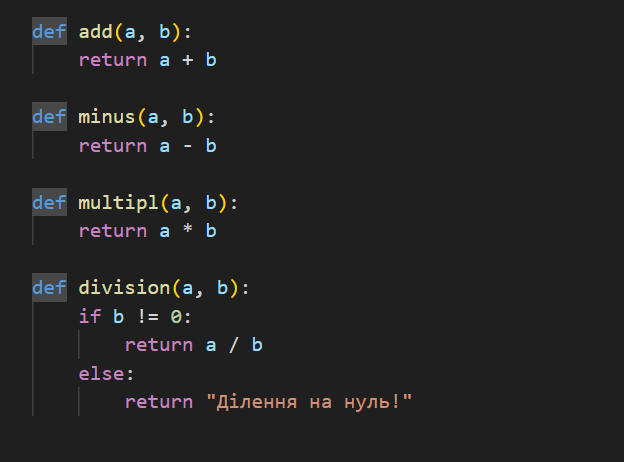


Виклик функцій.

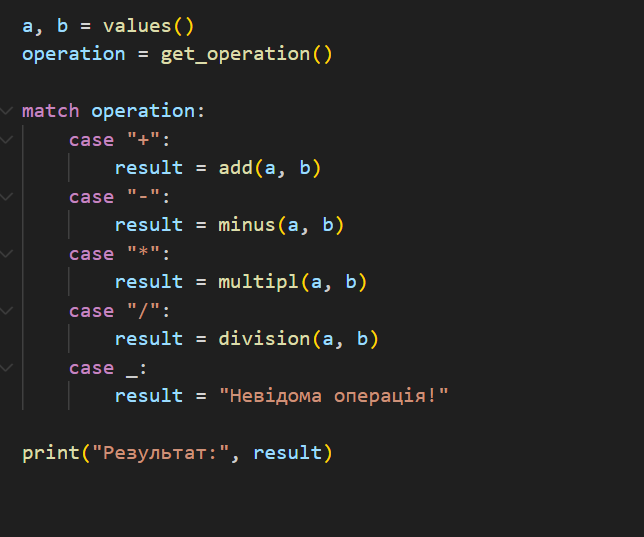
* Та сама програма калькулятор на основі оператора match.



Початок той самий що й в попередньому завданні, вводимо значення та операцію.



Функції дій з числами.



Викликаємо функцію *values()*, яка повертає значення змінних та функцію *get\_operation()*, яка повертає обрану користувачем операцію. Вводимо оператора match, яка перевірятиме значення змінної operation та відповідно якомусь з випадків код йтиме у підходящу для себе умову.

Якщо значення кейсу +, то викликається функція з додаванням, якщо кейс з – - з викликом функції віднімання, кейс з \* з викликом функції множення та кейс з / - з викликом функції ділення. В усіх інших випадках буде помилка введення операції. Виведення результату.