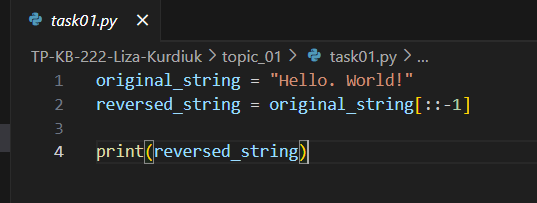
**Звіт про виконання практичних завдань до лекіїї з курсу Технологїї програмування на мові Python**

Зміст

1. **Звіт до теми №1……………………………………………………………3**
   1. Завдання 1**……………………………………………………………3**
   2. Завданння 2**……………………………………………………………4**
   3. Завдання 3**……………………………………………………………5**
2. **Звіт до теми №2……………………………………………………………6**
   1. Завдання 1**……………………………………………………………6**
   2. Завдання 2**……………………………………………………………7**
   3. Завдання 3**……………………………………………………………8**
3. **Звіт до теми №3……………………………………………………………10**
   1. Завдання 1**……………………………………………………………10**
   2. Завдання 2**……………………………………………………………11**
   3. Завдання 3**……………………………………………………………12**
   4. Завдання 4**……………………………………………………………13**
4. **Звіт до теми №4……………………………………………………………15**
   1. Завдання 1 **……………………………………………………………15**
   2. Завдання 2**……………………………………………………………16**
   3. Завдання 3**……………………………………………………………17**
5. **Звіт до теми №5……………………………………………………………19**
   1. Завдання 1 **……………………………………………………………19**
   2. Завдання 2 **……………………………………………………………20**
   3. Завдання 3 **……………………………………………………………21**
6. **Звіт до теми №6……………………………………………………………24**
   1. Завдання 1**……………………………………………………………24**
   2. Завдання 2**……………………………………………………………27**
7. **Звіт до теми №7……………………………………………………………29**
   1. Завдання 1 **……………………………………………………………29**
   2. Завдання 2 **……………………………………………………………29**
   3. Завдання 3 **……………………………………………………………31**
   4. Завдання 4 **……………………………………………………………32**

**Звіт до теми №1: Функції та змінні**

**Завдання 1 :** «Повернути рядок в зворотньому порядку»

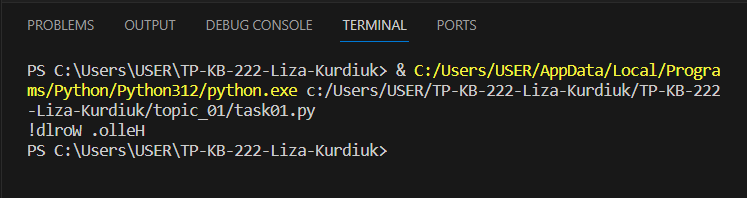


‘Original\_string[: :=-1]’ - Це синтаксис зрізу рядка. **-1** вказує на зрізування з кроком -1, що призводить до перевертання рядка

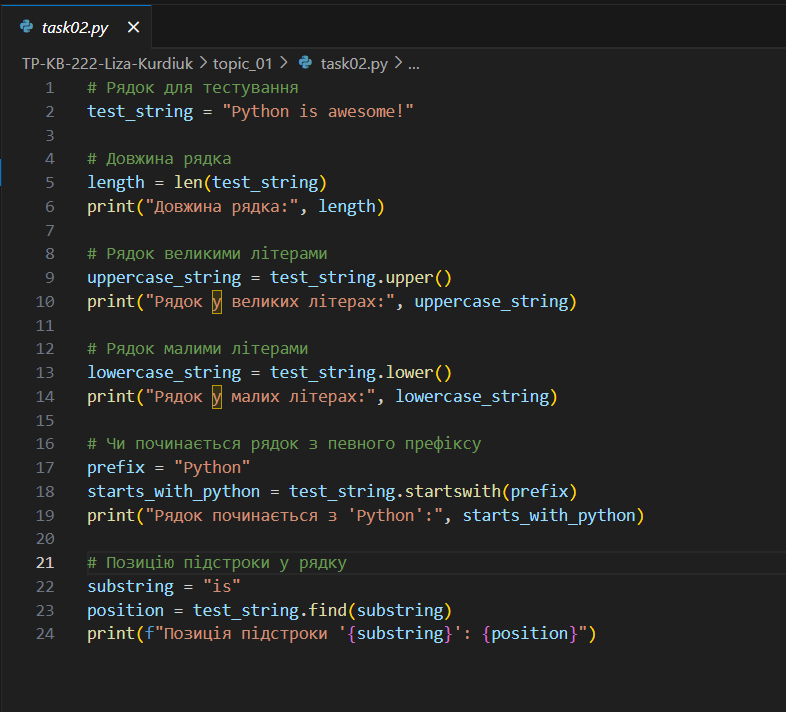
Результат присвоюється змінній ‘reserved\_string’

‘reserved\_string’ - Виводить результат на екран

Результат:

****

**Завдання 2: «**Протестувати базові функції для рядків**»**

****

**len(test\_string):** Повертає довжину рядка

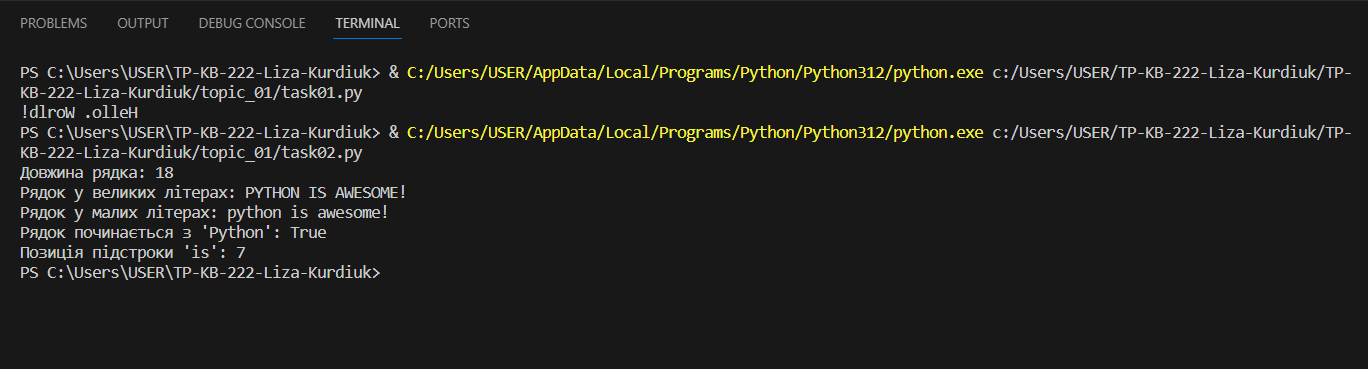
**upper():** Переводить рядок в верхній регістр.

**lower():** Переводить рядок в нижній регістр.

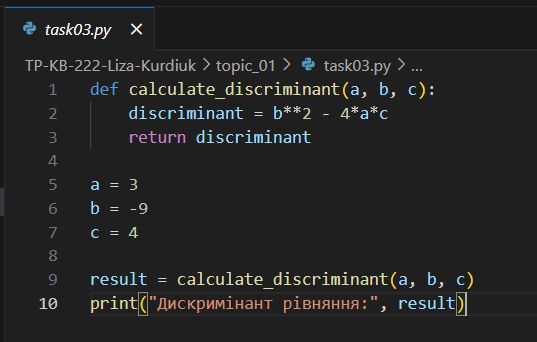
**Startwith(prefix):** використовується для перевірки того, чи рядок починається з певної підстроки. True, якщо рядок починається з вказаної підстроки, і False, якщо ні.

**Find(substring):** для знаходження позиції (індексу) першого входження підстроки (substring) урядку. Якщо підстрока не знайдена, то функція повертає -1.

**Результат:**

****

**Завдання 3: «**Написати функцію пошуку Дискримінанту**»**

****

В першому рядку ми визначаємо функцію **calculate\_discriminant**, яка приймає три аргументи - коефіцієнти a, b і c.

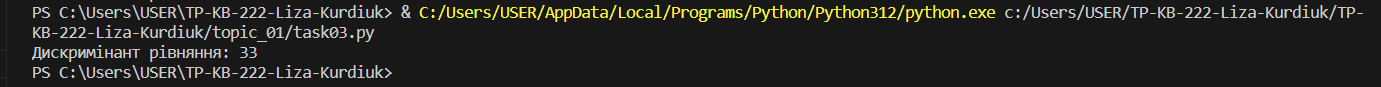
В другому: формула дискримінанта D = b^2 - 4ac.

Повернення дискримінанта

Визначаються значення коефіцієнтів a, b і c, і функція **calculate\_discriminant** викликається з цими значеннями.

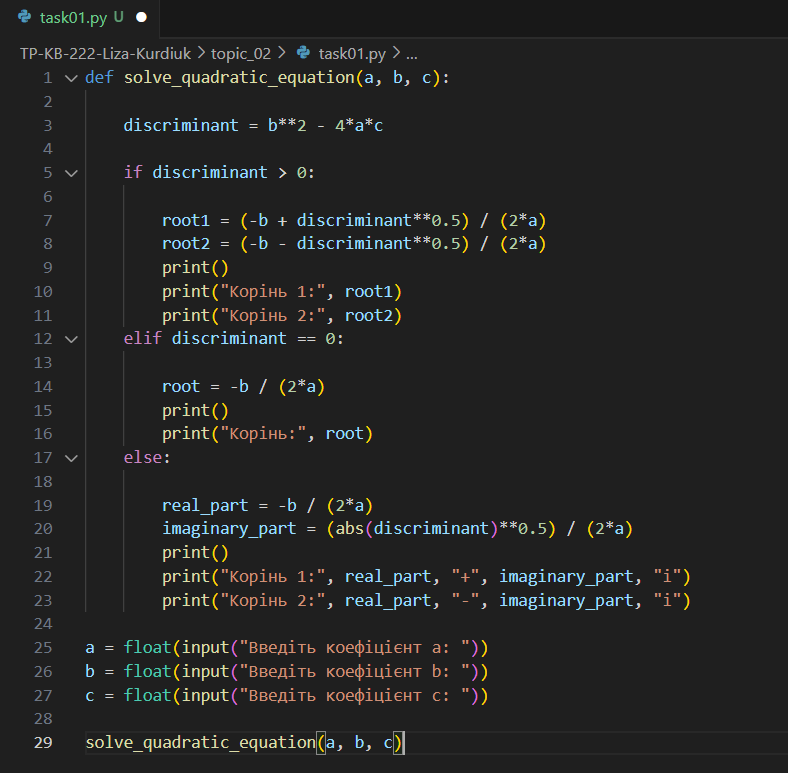
**Виведення результату:**

**Результат:**

****

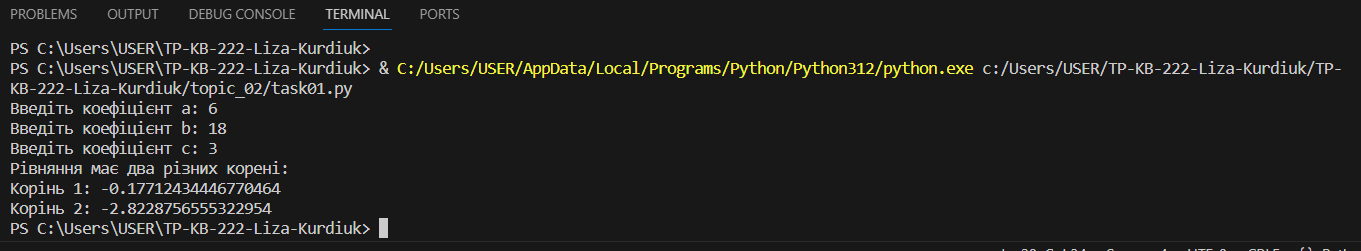
**Звіт до теми №2: Умовні переходи**

**Завдання 1:** Виконати пошуку коренів кввдратного рівняння враховуючи значення дискримінанту



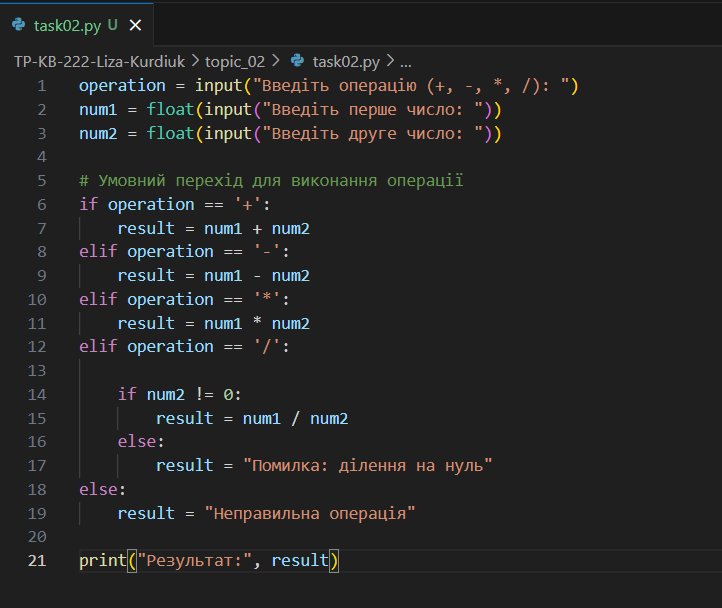
Ця функція приймає коефіцієнти a, b і c квадратного рівняння, обчислює дискримінант і виводить корені на екран в залежності від його значення.

Результат:



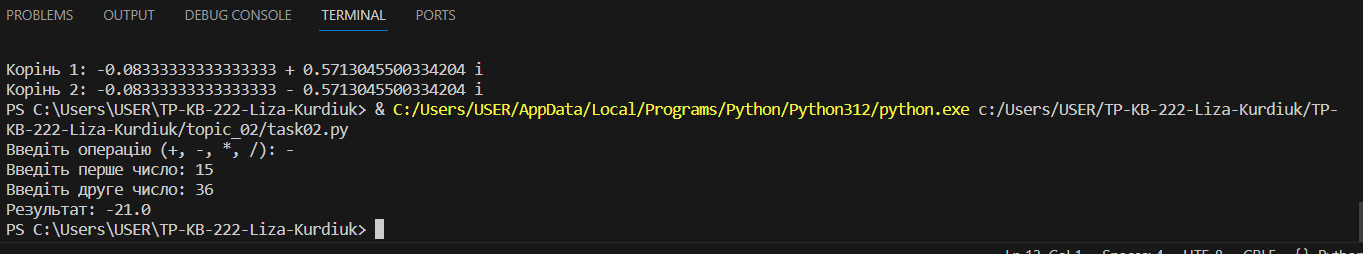
**Завдання 2:** Програма калькулятор на основі if elif else

Код:



Ця функція приймає два числа і операцію, виконує вибрану операцію та повертає результат.

Результат:

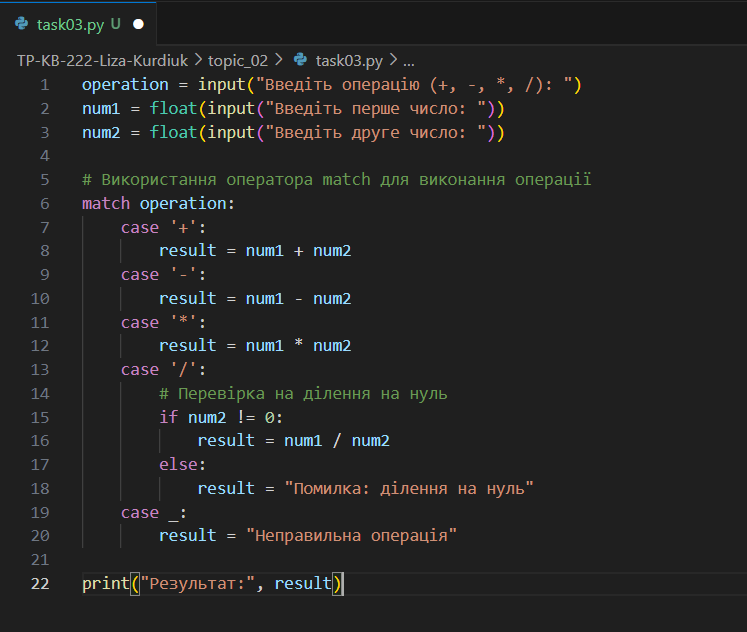


**Завдання 3**: Та сама програма калькулятор на основі оператора match

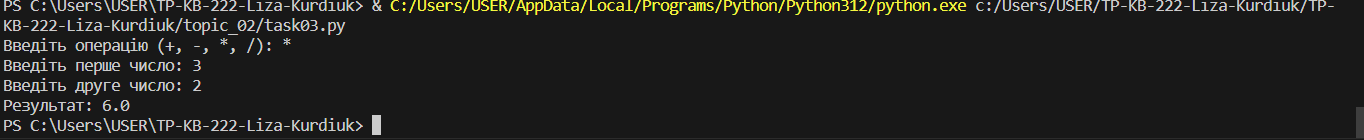
( Напиши код і пояснення)

Загалом ця функція аналогічна попередній, але використовує оператор **match** для визначення операції.

Код:



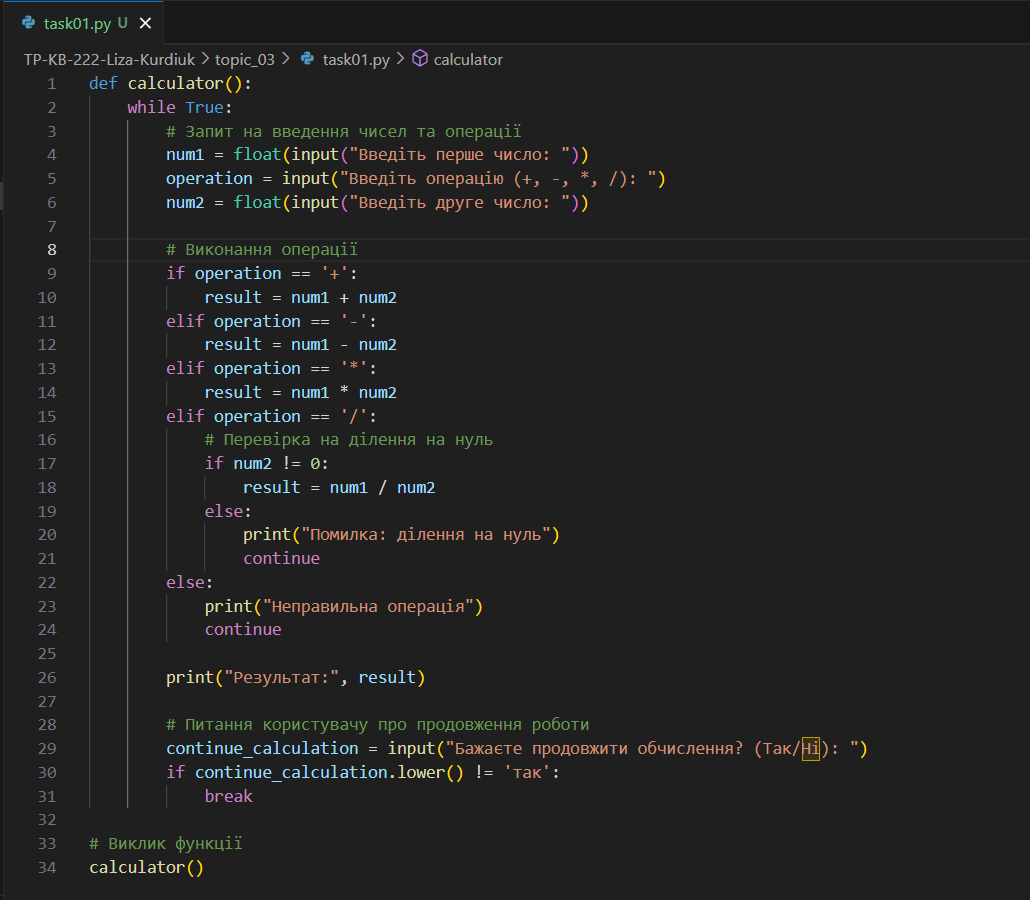
Результат:



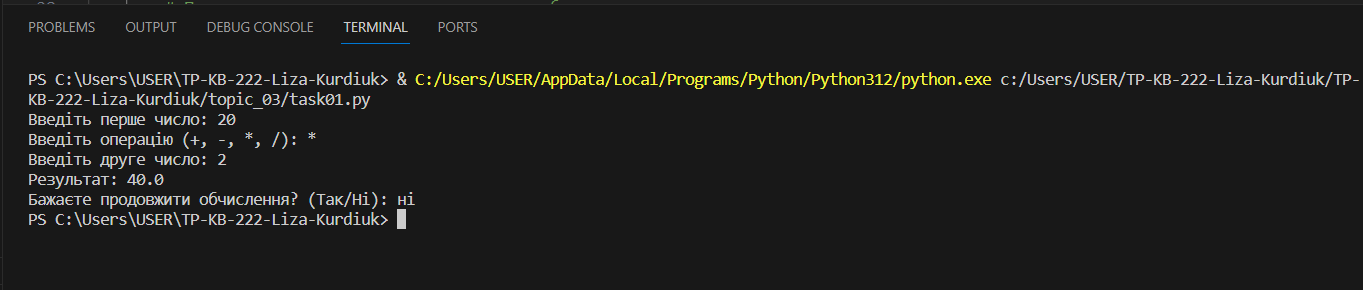
**Звіт до теми №3: Цикли**

**Завдання 1:** Написати програму калькулятор з постійними запитами на введення нових даних та операцій.

Ця програма працює в безкінечному циклі, де користувач може вводити числа та операції. Результат виводиться, і користувач може вирішити, чи продовжувати обчислення.



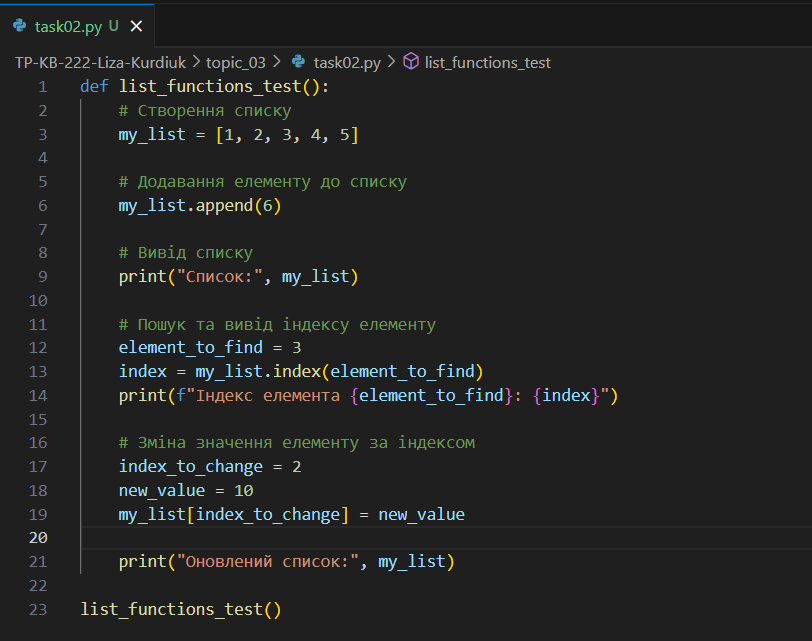
Результат:



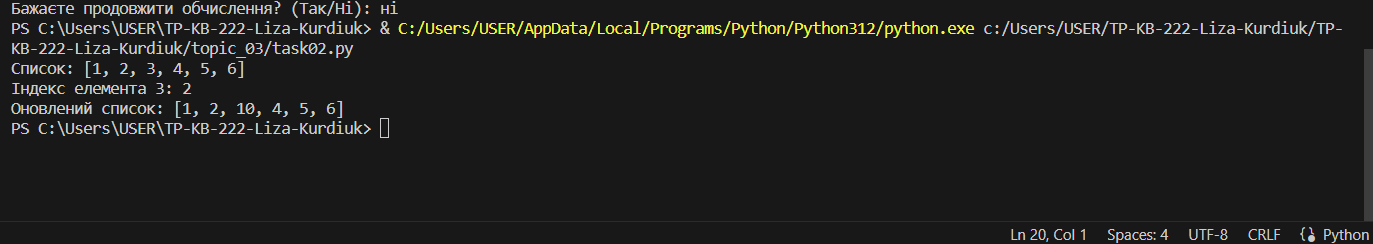
**Завдання 2:** Написати програму тестування функцій списків

Ця програма створює список, додає елемент, знаходить індекс елемента, змінює значення та виводить оновлений список.

Код:



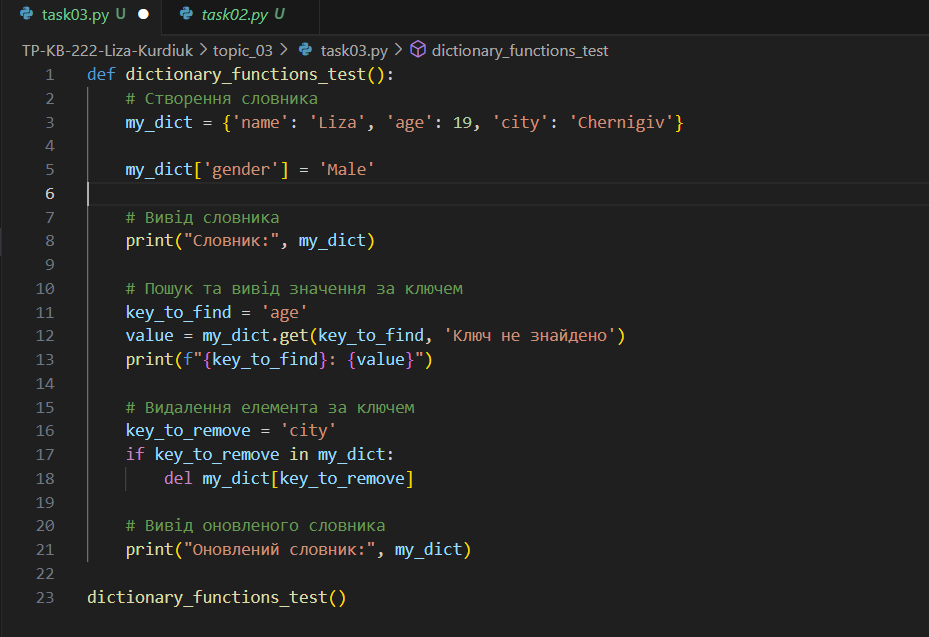
Результат:



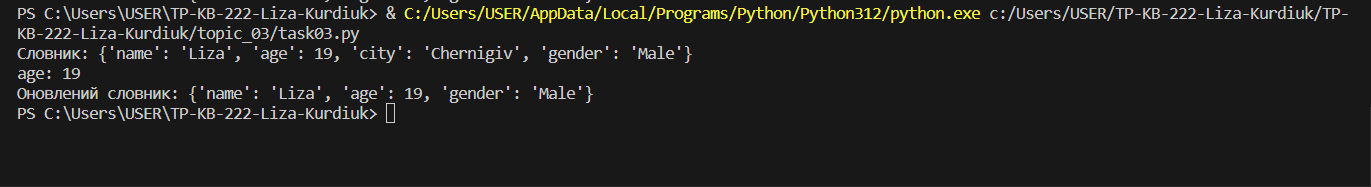
**Завдання 3:** Написати програму тестування функцій словників

Ця програма створює словник, додає новий ключ, знаходить та виводить значення за ключем, видаляє елемент за ключем та виводить оновлений словник.

Код:



Результат:

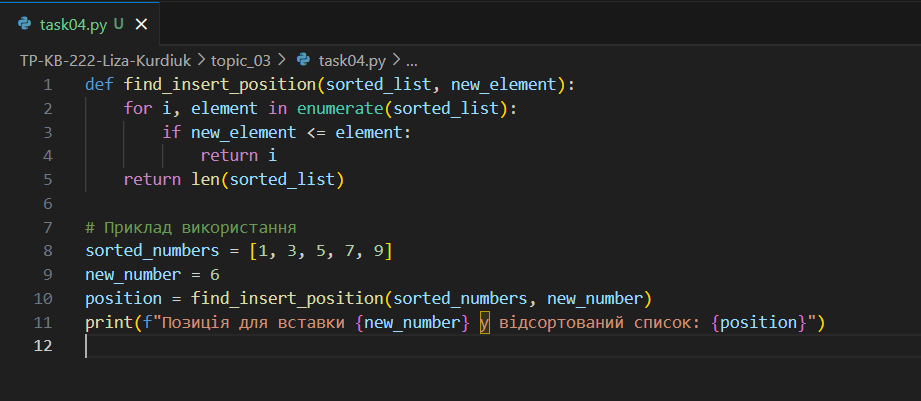


**Завдання 4:**  Написати функцію пошуку позиції для вставки нового елементу у відсортований список.

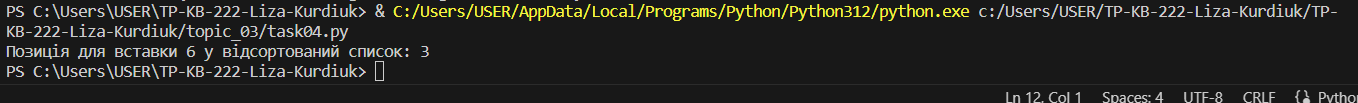
Ця функція приймає відсортований список і новий елемент, а потім знаходить позицію, на яку слід вставити новий елемент у відсортований список, зберігаючи впорядкування. Функція використовує цикл **for** та **enumerate**, щоб переглядати елементи списку та порівнювати їх із новим елементом. Якщо новий елемент менший або рівний поточному, функція повертає позицію. Якщо новий елемент більший за всі існуючі, то він має бути вставлений на кінець списку.

Наприклад, якщо маємо відсортований список **[1, 3, 5, 7, 9]** і хочемо вставити новий елемент **6**, функція поверне позицію **3**, оскільки **6** слід вставити між **5** і **7** для збереження порядку.

Код:



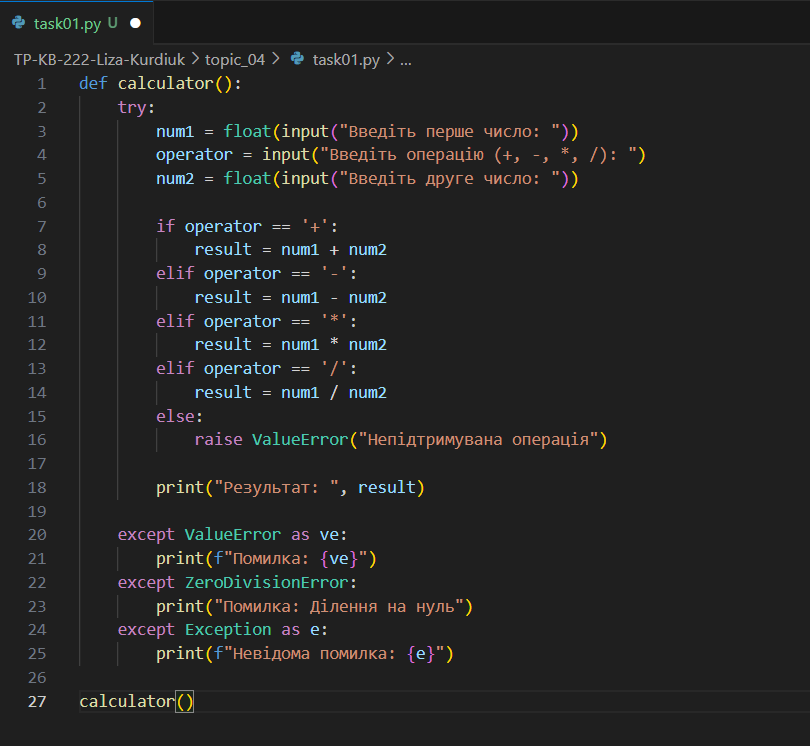
Результат:



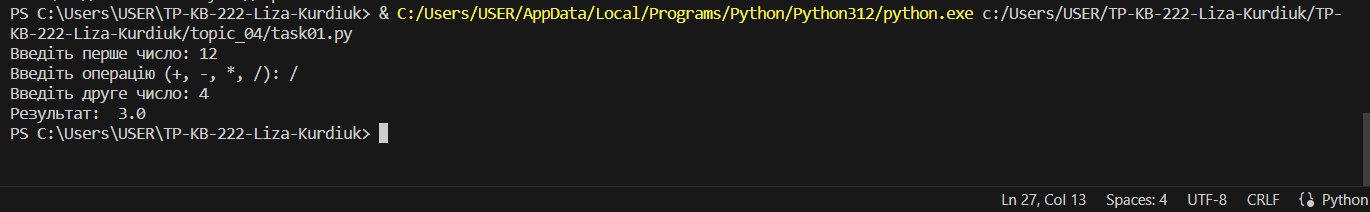
**Звіт до теми №4: Виняткові ситуації**

**Завдання 1:** Розширити програму калькулятор функцією запитів від користувача, що обробляє виняткові ситуації.

Код:

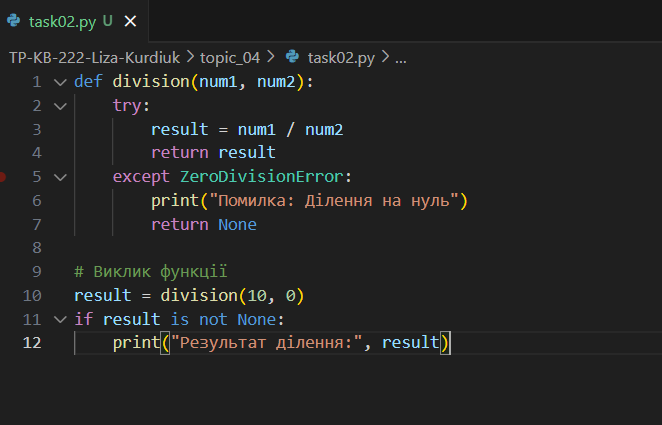


Результат:

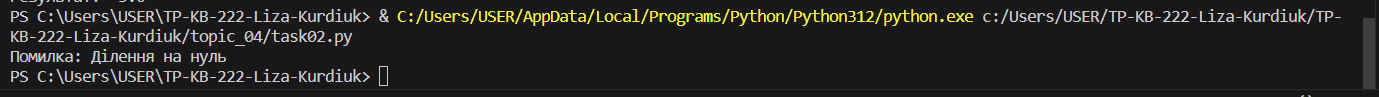


**Завдання 2 :** Розширити функцію ділення обробкою виняткової ситуації ділення но нуль

Код:



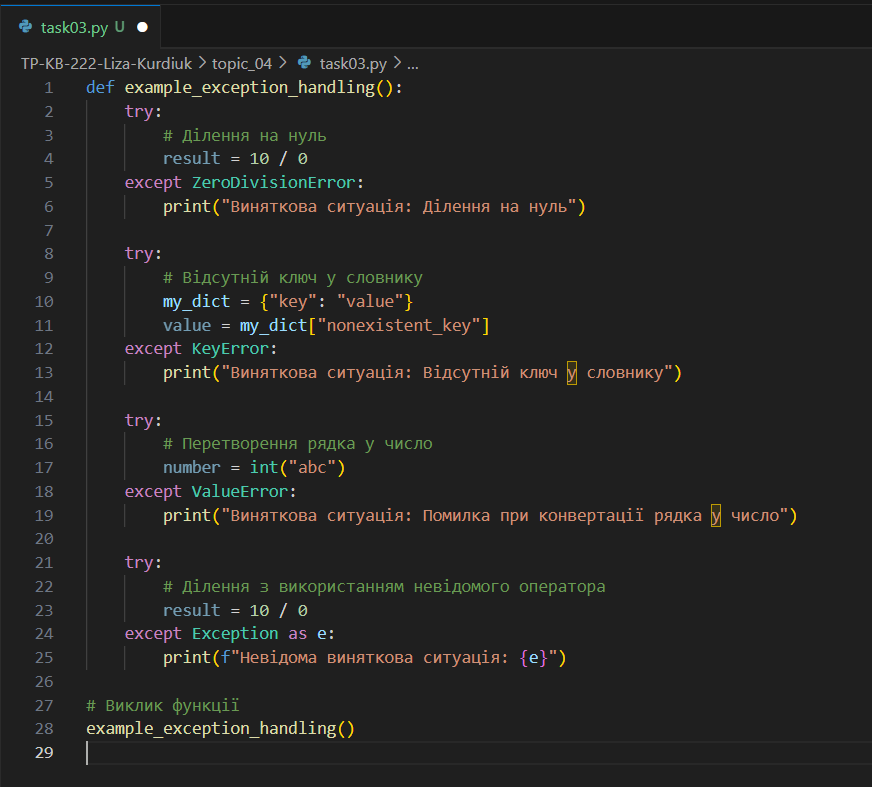
Рузультат:



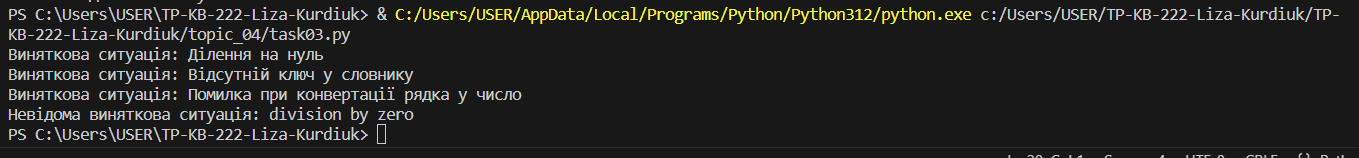
**Завдання 3:**  Ознайомитись зі списком виняткових ситуацій за посиланням

У цьому прикладі я використала різні виняткові ситуації, такі як **ZeroDivisionError**, **KeyError**, **ValueError**, та загальний клас **Exception**. Кожна ситуація виводить відповідне повідомлення при виникненні помилки.

Код:



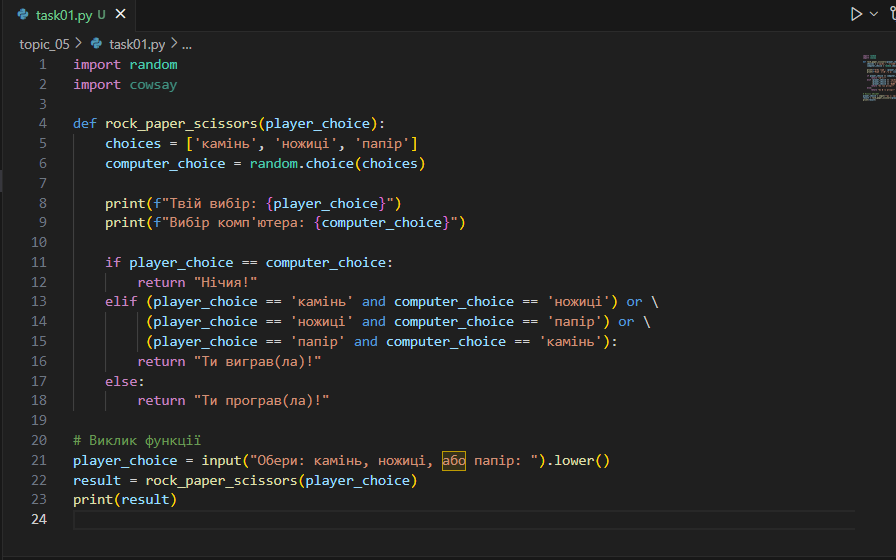
Результат:



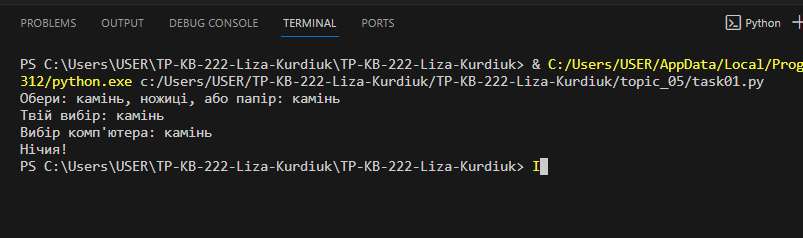
**Звіт до теми №5: Бібліотеки**

**Завдання 1:** Гра Камінь Ножиці Папер

Код:

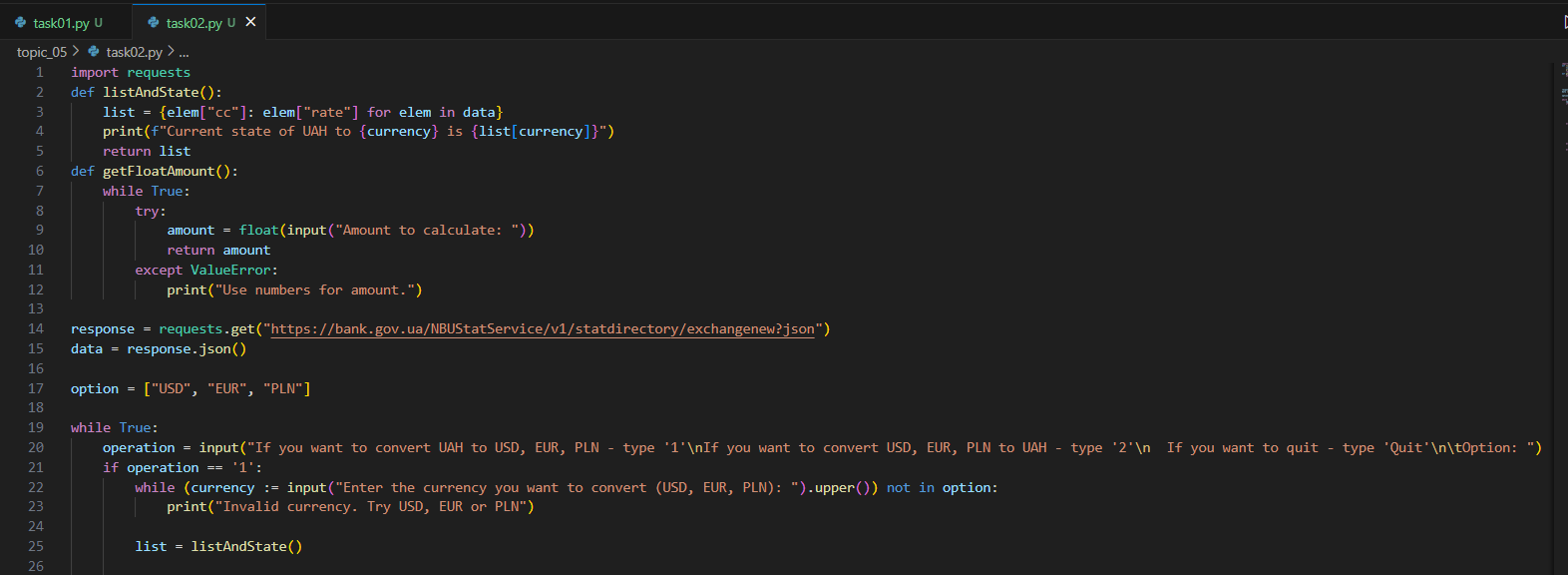


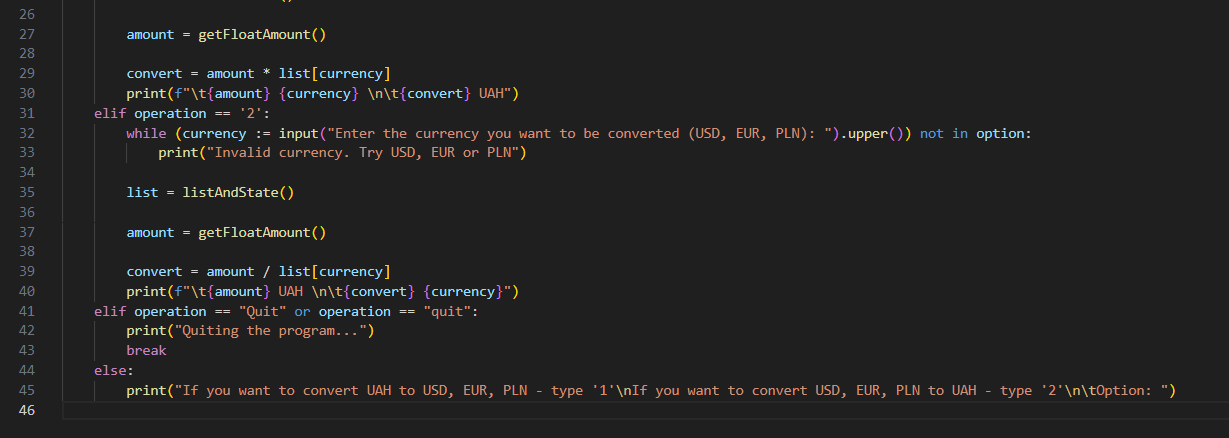
Результат:



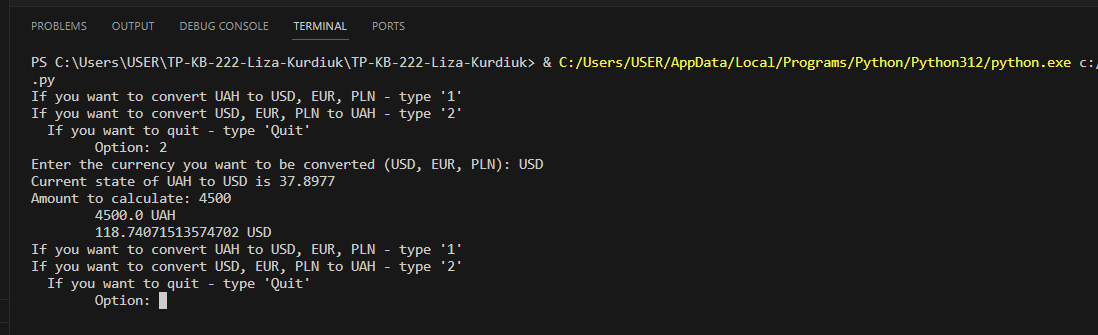
**Завдання 2: Конвертор валют**

Код:





Результат:



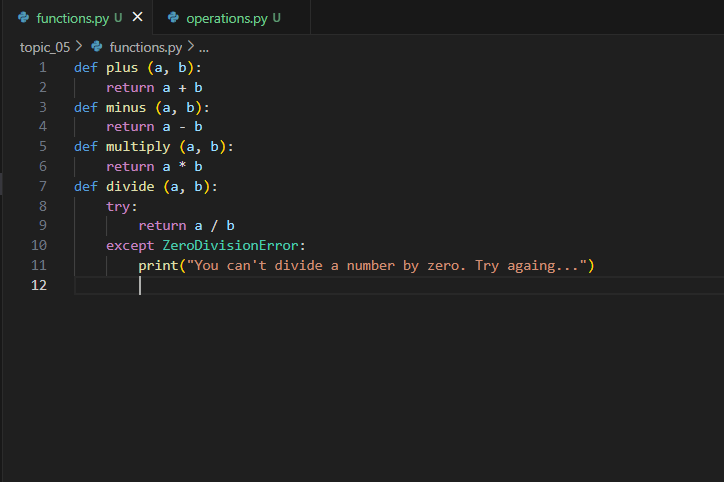
**Завдання 3:** Модулі для програми калькулятор

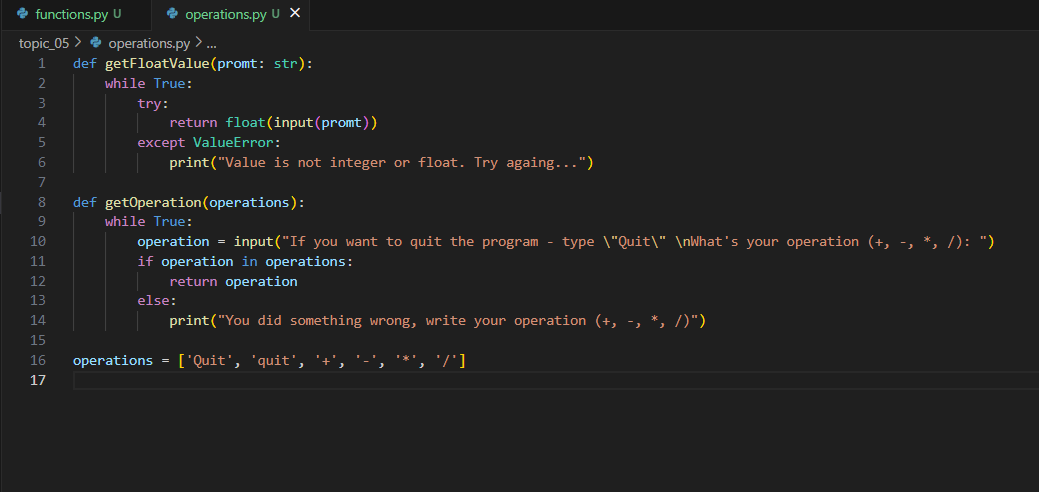
Щоб реалізувати код спочатку перемістила функції додавання, віднімання, множення та ділення до файлу functions.py.

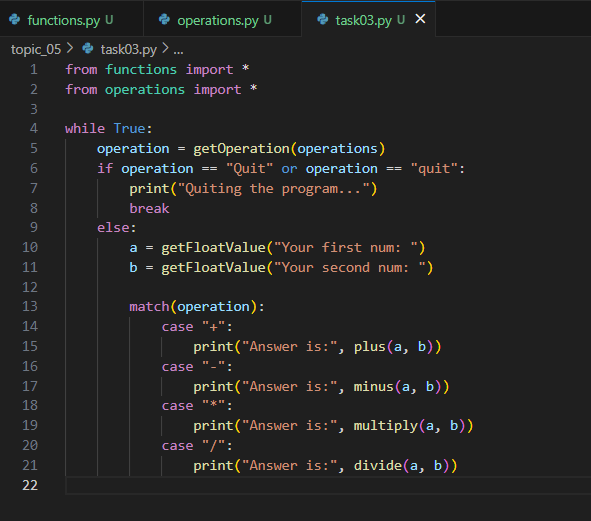
Також потім виділила функції, які відповідають за введення даних для операцій та сами операції, у файл operations.py.

Основний код помістила в task03.py

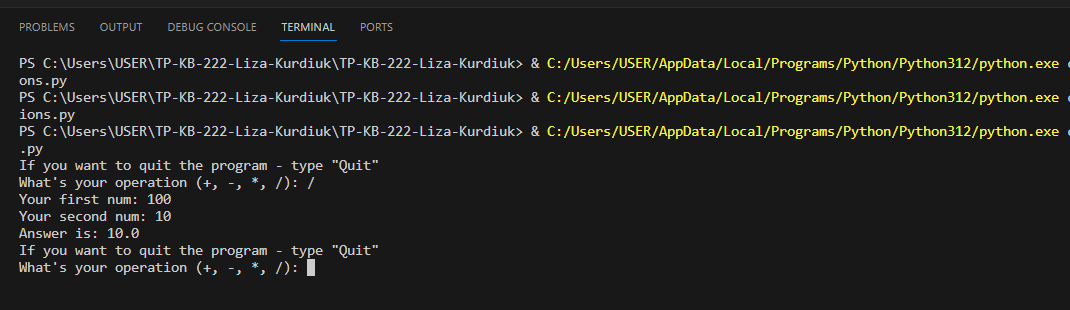
Код:







Результат:



**Звіт до теми №6:Робота з файлами**

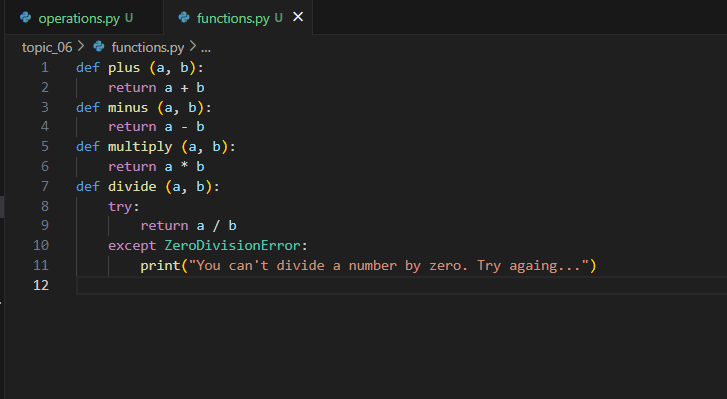
**Завдання 1:** Використання lambda функцій для функції сортування

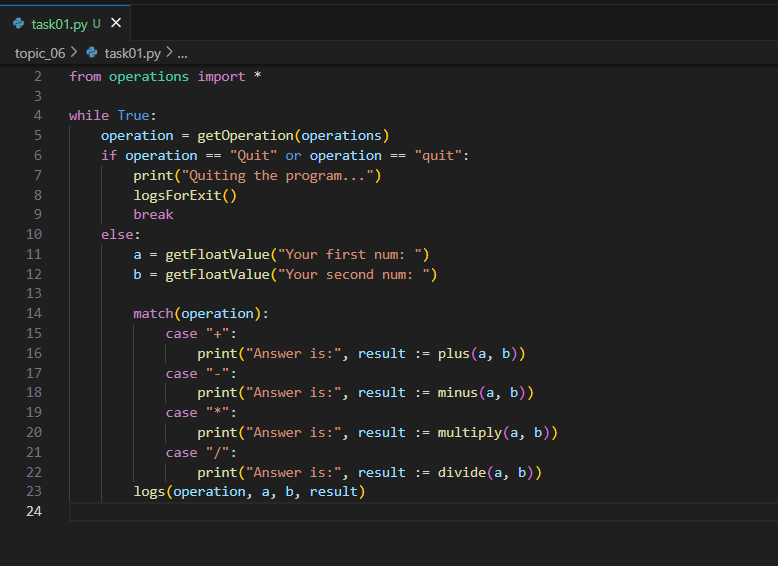
Код:

**lambda** - це анонімна функція в Python. Вона дозволяє створювати функції "на льоту", без задання імені. **lambda** може бути корисною, наприклад, при сортуванні списків, коли ви хочете вказати критерії сортування без визначення окремої функції.

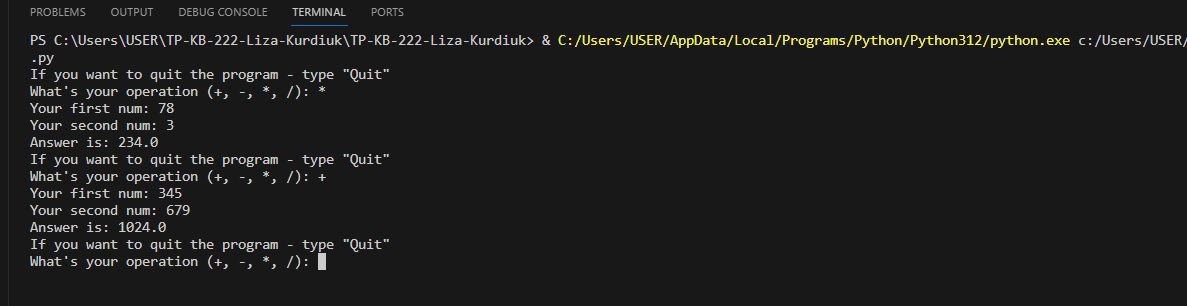
В файлі operations.py імпортував два модулі - datetime та os.



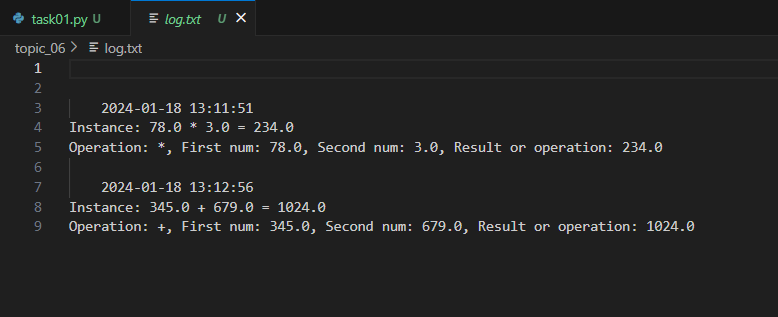




Результат:

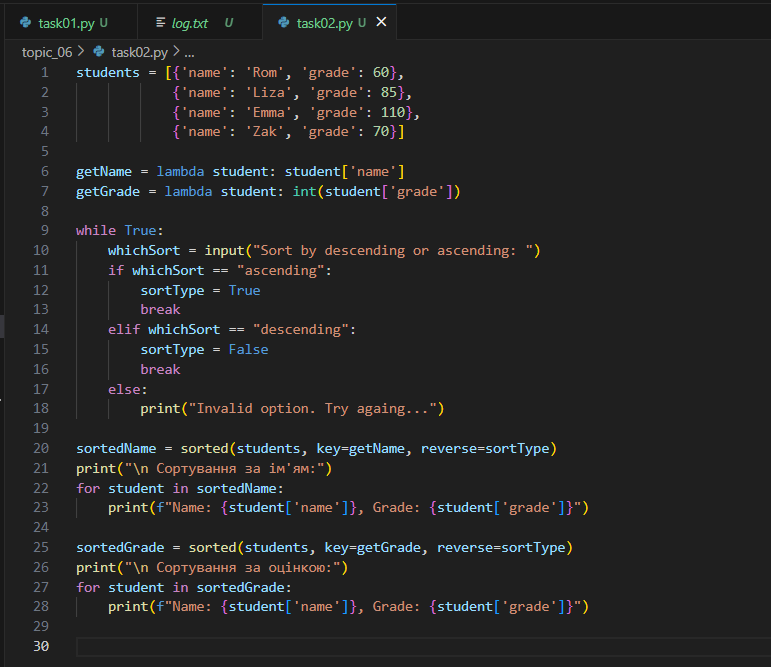


Після успішно виконаної операції, результат логувався

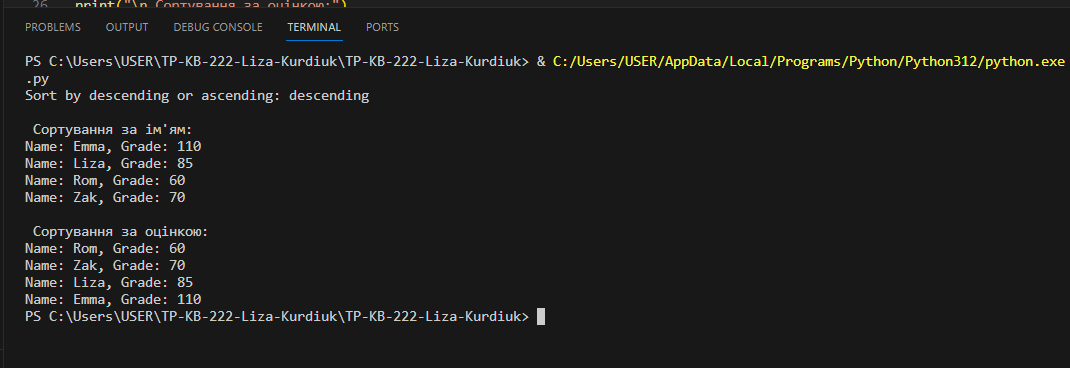


**Завдання 2:** Логування всіх дій в застосунку Калькулятор

Код:



Сортування відбувається так:



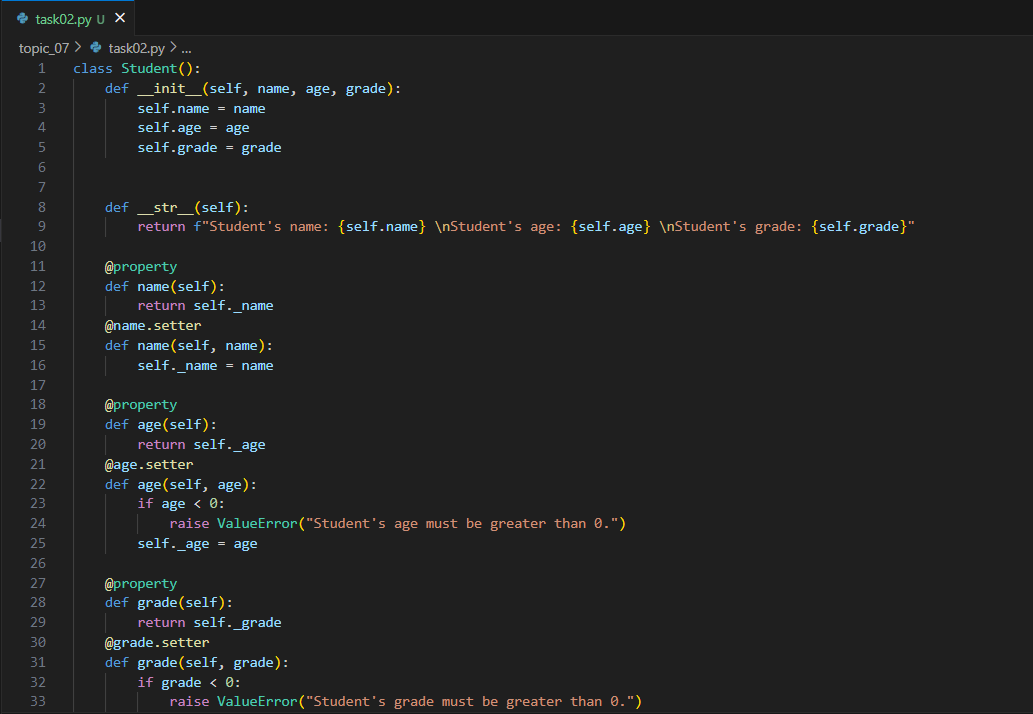
**Звіт до теми №7: ОПП**

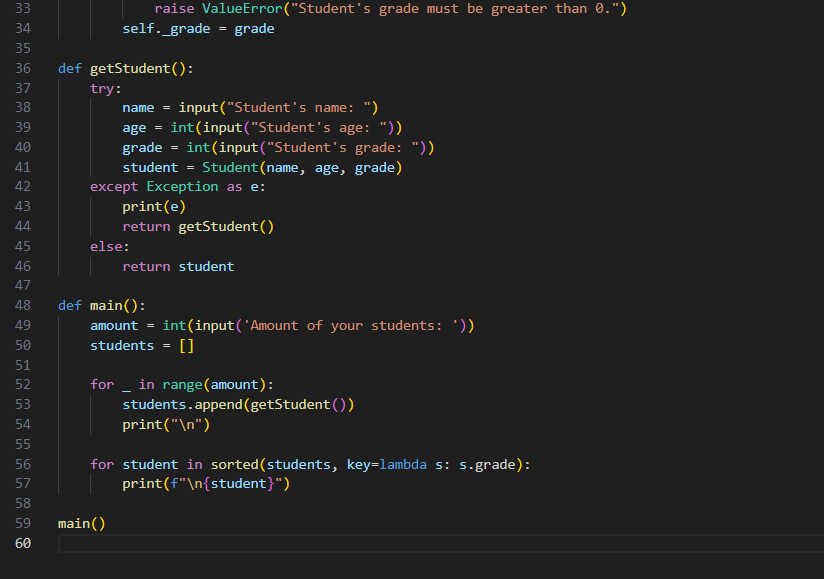
**Завдання 1 :** документація про Сlass

Документація про клас у програмуванні описує концепцію класу як шаблону для створення об'єктів. У програмуванні клас визначає атрибути (змінні) та методи (функції), які можна використовувати для створення конкретних об'єктів. Використання класів є основною складовою об'єктно-орієнтованого програмування (ООП).

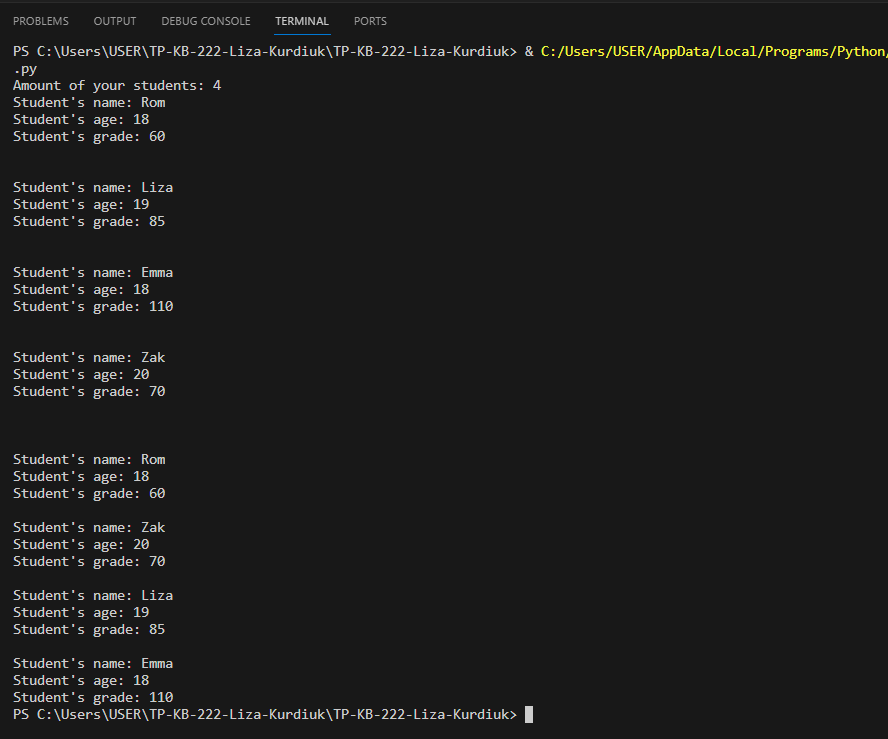
**Завдання 2: func**

Програма надає можливість працювати з об'єктами класу MyClass та виконує ряд корисних операцій,виведення відсортованого списку об'єктів за віком, а також можливість додавання нового об'єкта до списку та збереження його у файлі.



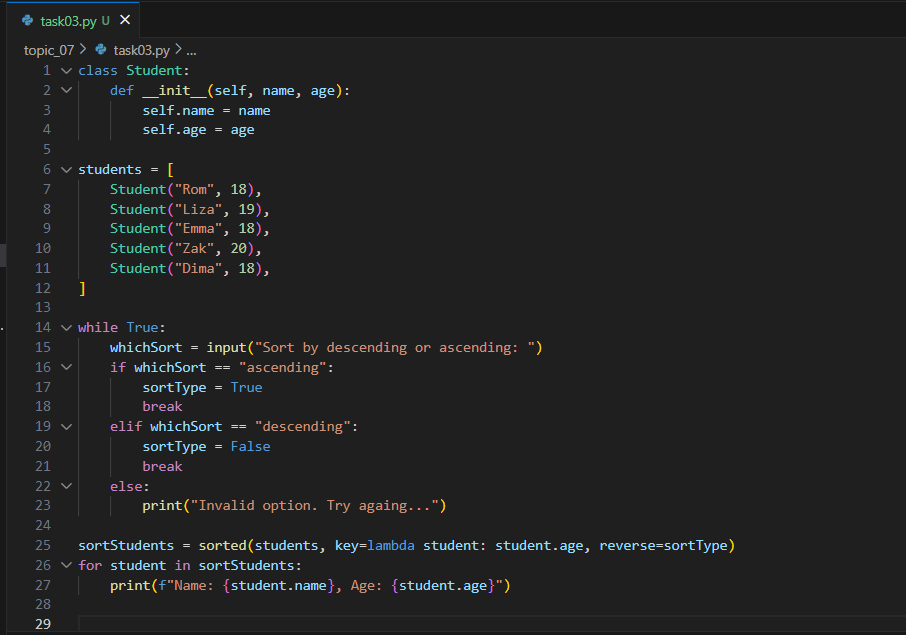


Результат:

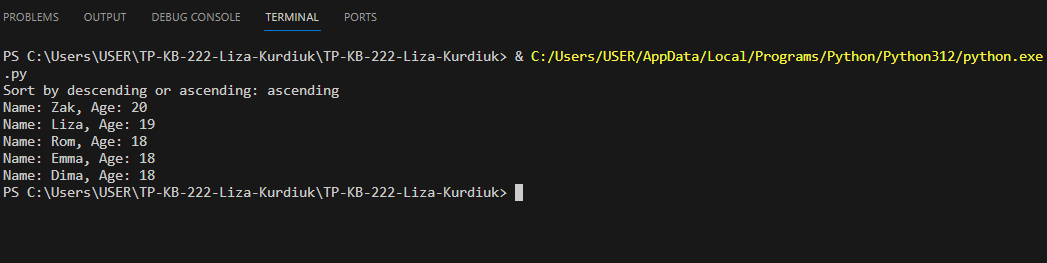


**Завдання3 : Calc with OOP**

В коді реалізовано клас Student з атрибутами name та age.



Результат:



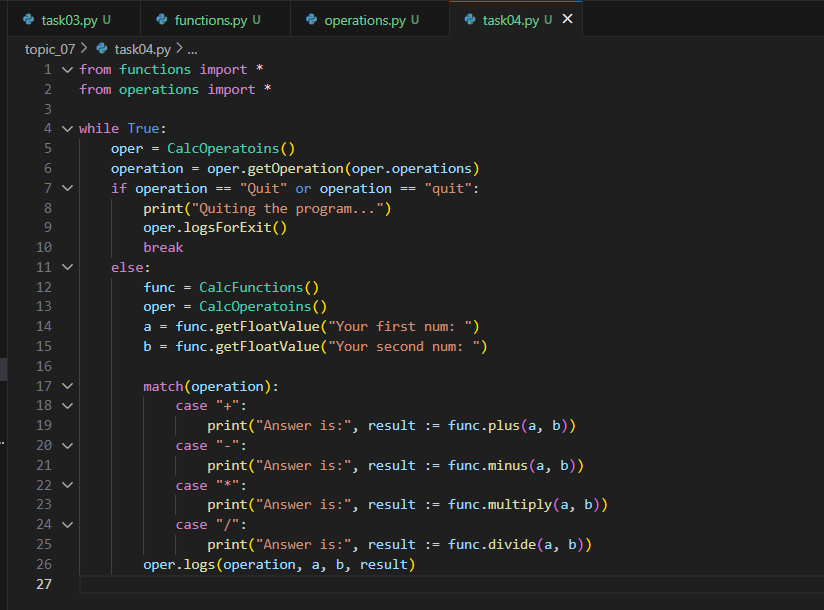
**Завдання4:** Calc with OOP

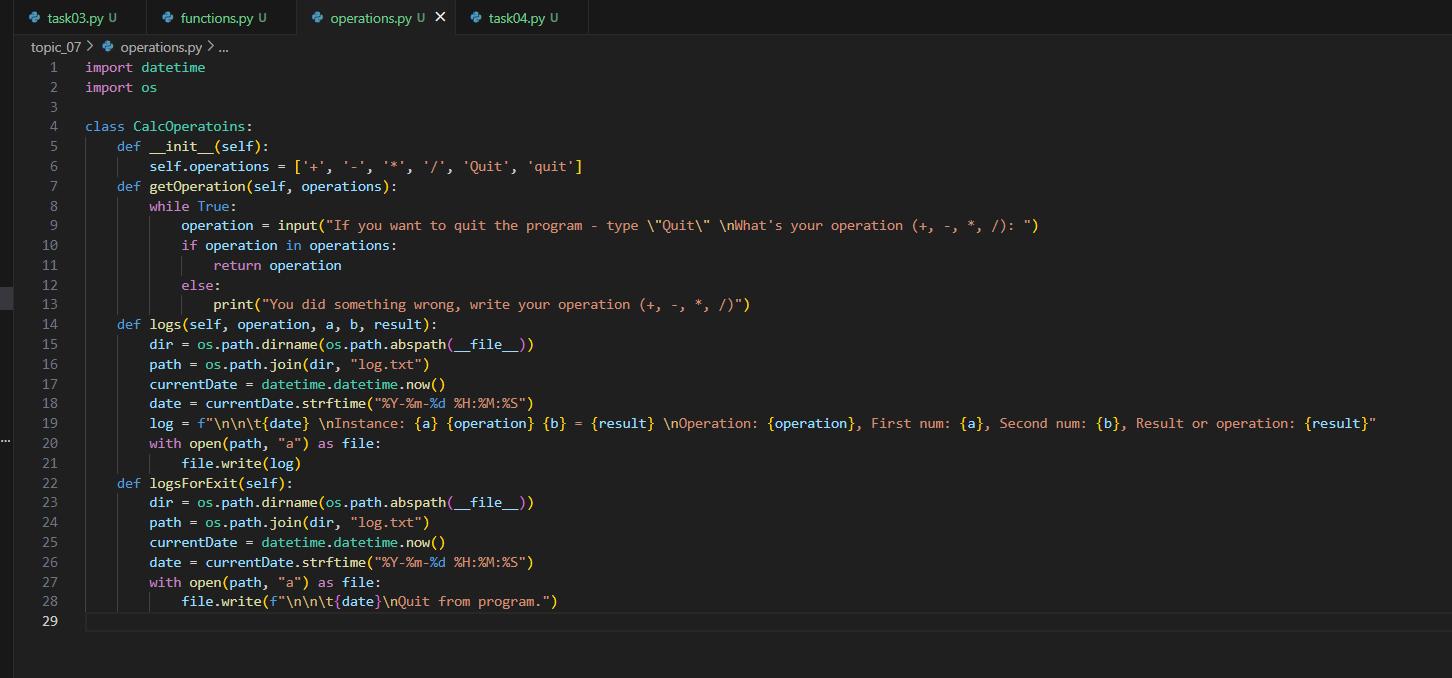
Застосувати принципи об'єктно-орієнтованого програмування для переписування програми Калькулятор. Задачу слід виконати з використанням модульного підходу.

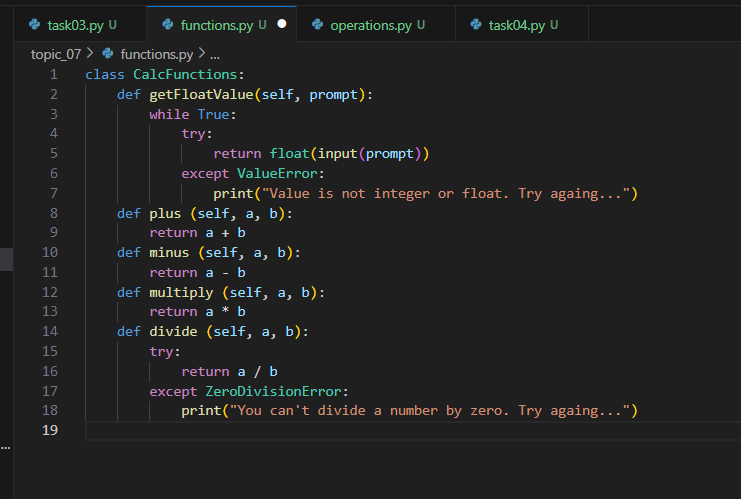
Переписала файл functions.py за принципами ООП.

Переписала файл operations.py за принципами ООП.

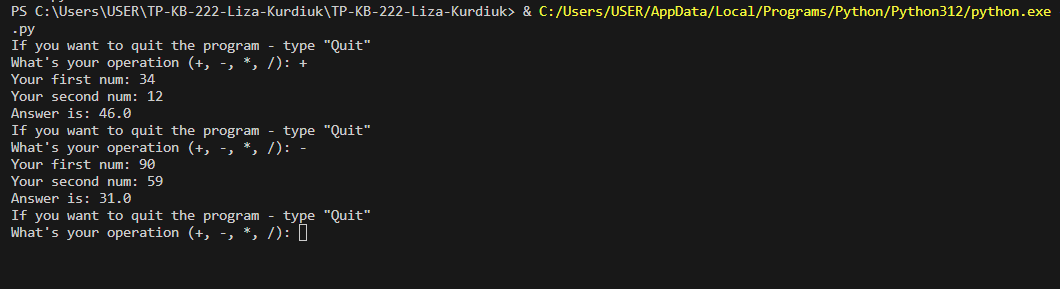
Код:







Результат:



Після успішного виконання файл результат логувався

