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

**Зміст**

[Завдання №1.1 3](#_Toc147598041)

[Хід виконання завдання: 3](#_Toc147598042)

[Текст програми: 3](#_Toc147598043)

[Посилання на github: 3](#_Toc147598044)

[Знімок екрану з посилання на github: 3](#_Toc147598045)

[Завдання №1.2 4](#_Toc147598046)

[Хід виконання завдання: 4](#_Toc147598047)

[Текст програми: 4](#_Toc147598048)

[Посилання на github: 4](#_Toc147598049)

[Знімок екрану з посилання на github: 4](#_Toc147598050)

[Завдання №1.3 6](#_Toc147598051)

[Хід виконання завдання: 6](#_Toc147598052)

[Текст програми: 6](#_Toc147598053)

[Посилання на github: 6](#_Toc147598054)

[Знімок екрану з посилання на github: 6](#_Toc147598055)

[Завдання №2.1 7](#_Toc147598056)

[Хід виконання завдання: 7](#_Toc147598057)

[Текст програми: 7](#_Toc147598058)

[Посилання на github: 7](#_Toc147598059)

[Знімок екрану з посилання на github: 7](#_Toc147598060)

[Завдання №2.2 9](#_Toc147598061)

[Хід виконання завдання: 9](#_Toc147598062)

[Текст програми: 9](#_Toc147598063)

[Посилання на github: 9](#_Toc147598064)

[Знімок екрану з посилання на github: 9](#_Toc147598065)

[Завдання №2.3 11](#_Toc147598066)

[Хід виконання завдання: 11](#_Toc147598067)

[Текст програми: 11](#_Toc147598068)

[Посилання на github: 11](#_Toc147598069)

[Знімок екрану з посилання на github: 11](#_Toc147598070)

[Завдання №3.1 13](#_Toc147598071)

[Хід виконання завдання: 13](#_Toc147598072)

[Текст програми: 13](#_Toc147598073)

[Посилання на github: 14](#_Toc147598074)

[Знімок екрану з посилання на github: 14](#_Toc147598075)

[Завдання №3.2 15](#_Toc147598076)

[Хід виконання завдання: 15](#_Toc147598077)

[Текст програми: 15](#_Toc147598078)

[Посилання на github: 16](#_Toc147598079)

[Знімок екрану з посилання на github: 16](#_Toc147598080)

[Завдання №3.3 17](#_Toc147598081)

[Хід виконання завдання: 17](#_Toc147598082)

[Текст програми: 17](#_Toc147598083)

[Посилання на github: 17](#_Toc147598084)

[Знімок екрану з посилання на github: 17](#_Toc147598085)

[Завдання №3.4 18](#_Toc147598086)

[Хід виконання завдання: 18](#_Toc147598087)

[Текст програми: 18](#_Toc147598088)

[Посилання на github: 18](#_Toc147598089)

[Знімок екрану з посилання на github: 18](#_Toc147598090)

[Завдання №4.1-2 19](#_Toc147598091)

[Хід виконання завдання: 19](#_Toc147598092)

[Текст програми: 19](#_Toc147598093)

[Посилання на github: 20](#_Toc147598094)

[Знімок екрану з посилання на github: 20](#_Toc147598095)

[Завдання №4.3 22](#_Toc147598096)

[Хід виконання завдання: 22](#_Toc147598097)

[Текст програми: 22](#_Toc147598098)

[Посилання на github: 23](#_Toc147598099)

[Знімок екрану з посилання на github: 23](#_Toc147598100)

Звіт до Теми №1

Функції та змінні

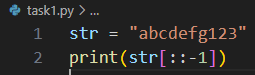
Завдання №1.1

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

Хід виконання завдання:

1. Вписав необхідний текст.
2. Виконав дії для перетворення рядку задом наперед.
3. Виконав програму та перевірив її на успішність.

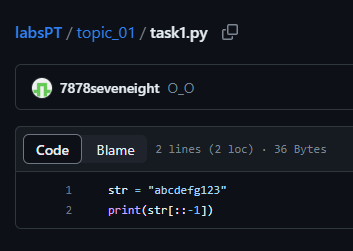
Текст програми:



Посилання на github:

<https://github.com/7878seveneight/labsPT/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github:



Завдання №1.2

Перевірити роботу функцій strip() , capitalize(), title(), upper(), lower() та т.п.

Хід виконання завдання:

1. Зробив змінні для тексту на якому будуть проходити перевірки та написав сам текст.
2. Ввів змінні для зберігання в них форматованого тексту.
3. Надрукував форматований текст.

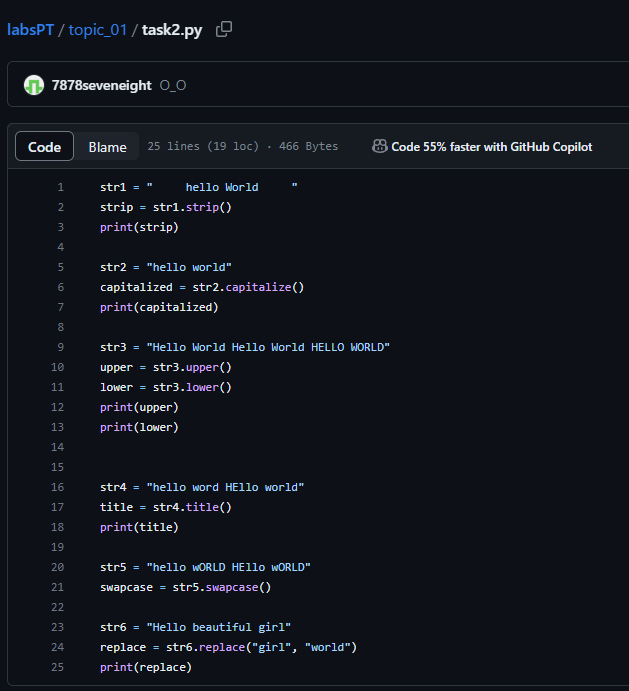
Текст програми:



Посилання на github:

<https://github.com/7878seveneight/labsPT/blob/main/topic_01/task2.py>

Знімок екрану з посилання на github:



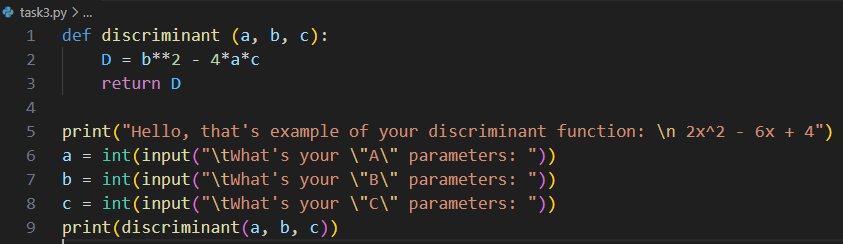
Завдання №1.3

Написати функцію для знаходження дискріминанта.

Хід виконання завдання:

1. Написав функцію, для знаходження дискріминта, з трьома аргументами.
2. Ввів змінні для підставлення аргументів у функцію.
3. Вивів відповідь на приклад.

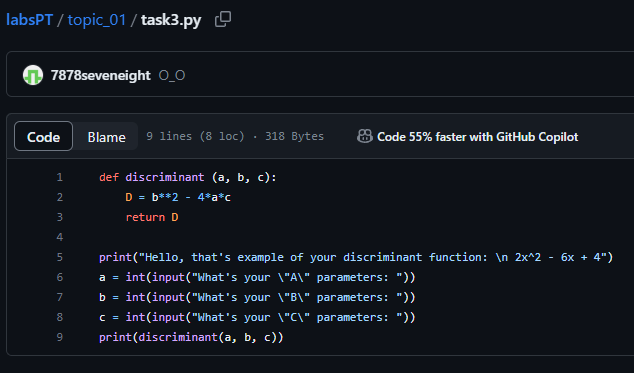
Текст програми:



Посилання на github:

<https://github.com/7878seveneight/labsPT/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github:



Звіт до Теми №2

Функції та змінні

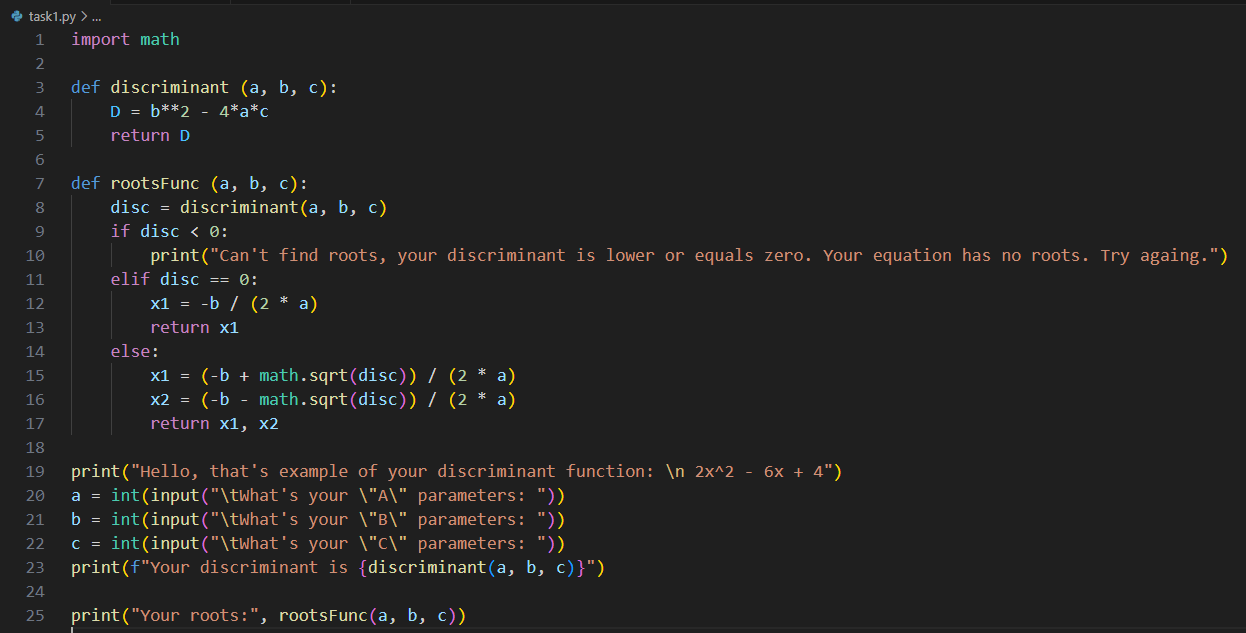
Завдання №2.1

Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Хід виконання завдання:

1. Імпортував бібліотеку math для знахоження квадратного кореня.
2. Вставив код функції дискимінанта з минулого завдання.
3. Написав функцію для знахоження коренів.
4. Дізнався значення змінних від користувача через input.
5. Підставив ці значення та вивів результати.

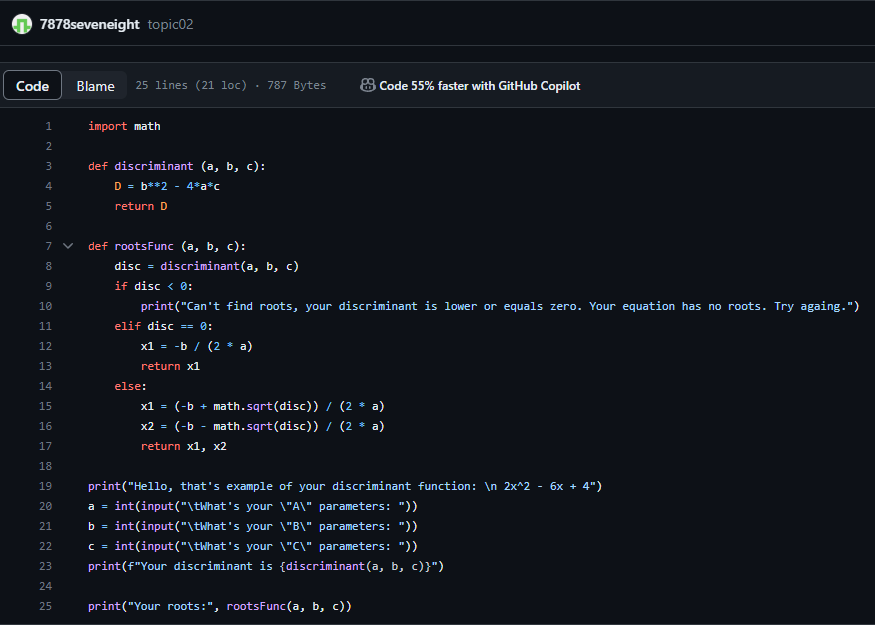
Текст програми:



Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_02/task1.py)

Знімок екрану з посилання на github:



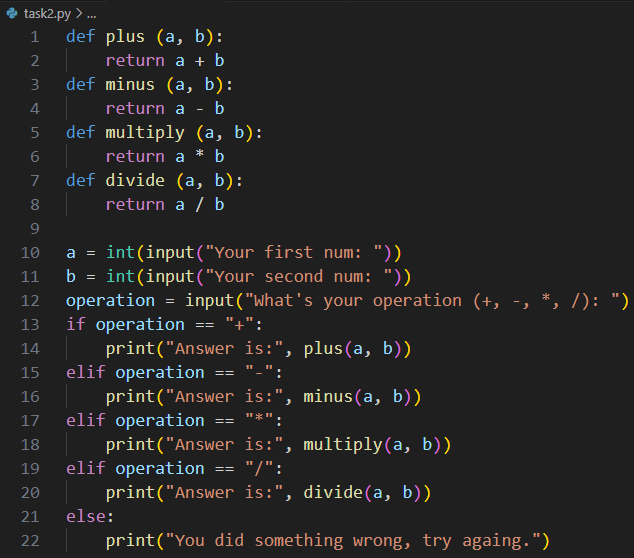
Завдання №2.2

Написати програму калькулятор використовуючи if else конструкцію.

Хід виконання завдання:

1. Написав функції з простими математичними діями.
2. Дізнався значення змінних двох чисел та операцію від користувача через input.
3. Написав логіку програми для підставлення чисел та операції використовуючи if else конструкцію, вивів результат.

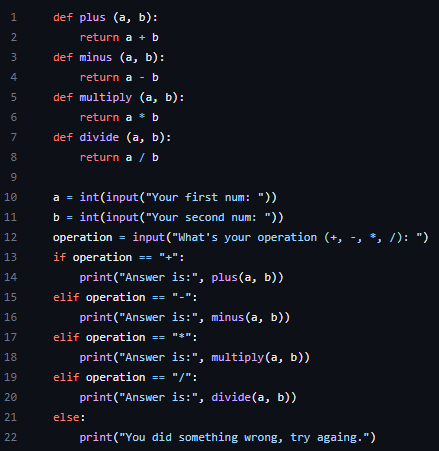
Текст програми:



Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_02/task2.py)

Знімок екрану з посилання на github:



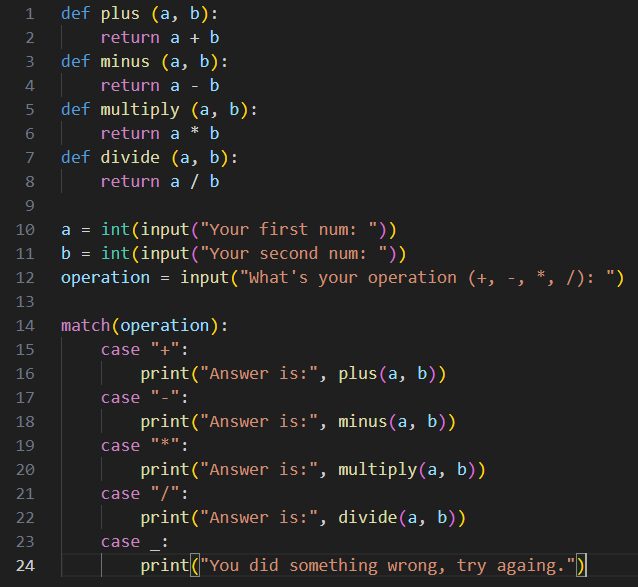
Завдання №2.3

Написати програму калькулятор використовуючи match конструкцію

Хід виконання завдання:

1. Скопіював код, математичних функцій та отримання зміних від користувача, дій зі завдання вище.
2. Написав логіку програми для підставлення чисел та операції використовуючи match конструкцію, вивів результат.

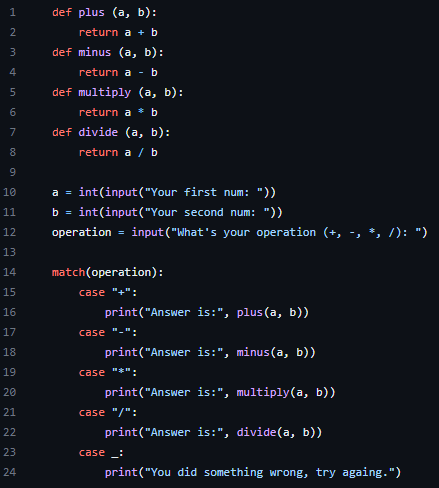
Текст програми:



Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_02/task3.py)

Знімок екрану з посилання на github:



Звіт до Теми №2

Функції та змінні

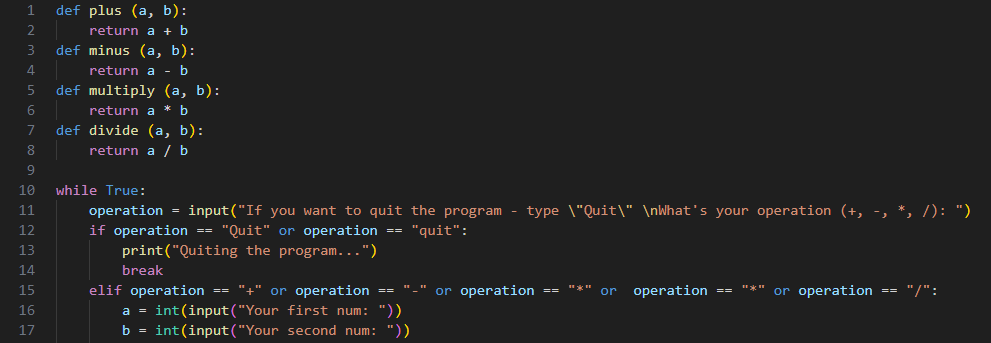
Завдання №3.1

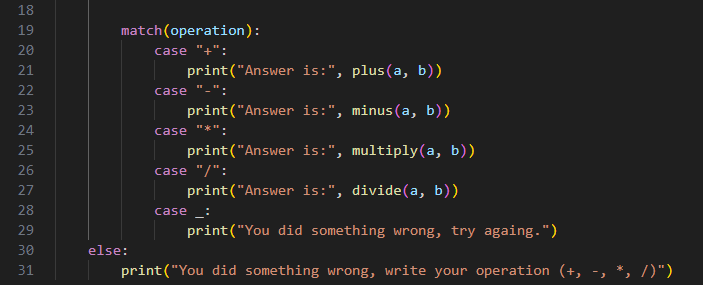
Переписати калькулятор з попередніх завдань використовуючи нескінченний цикл. Запитувати кожен раз дві функції та оператор. У випадку коли нажметься щось не те, додати можливість вийти.

Хід виконання завдання:

1. Скопіював функції та деякі деталі коду калькулятора з попередніх завдань.
2. Зробив нескінченний цикл, який запитує кожен раз дві функції та оператор.
3. У випадку коли нажметься щось не те, додав можливість виходу.

Текст програми:





Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_03/task1.py)

Знімок екрану з посилання на github:



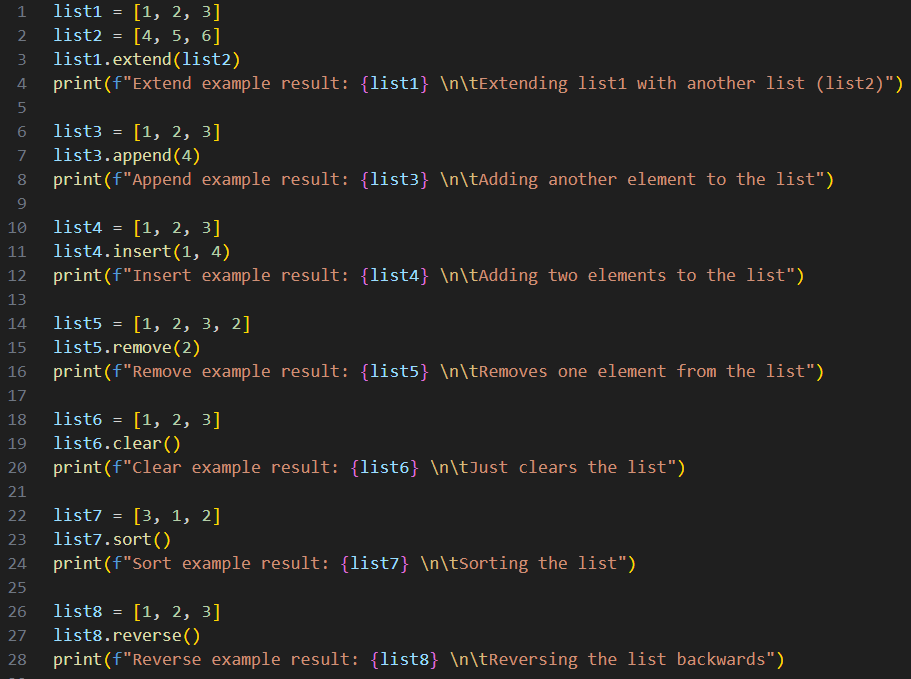
Завдання №3.2

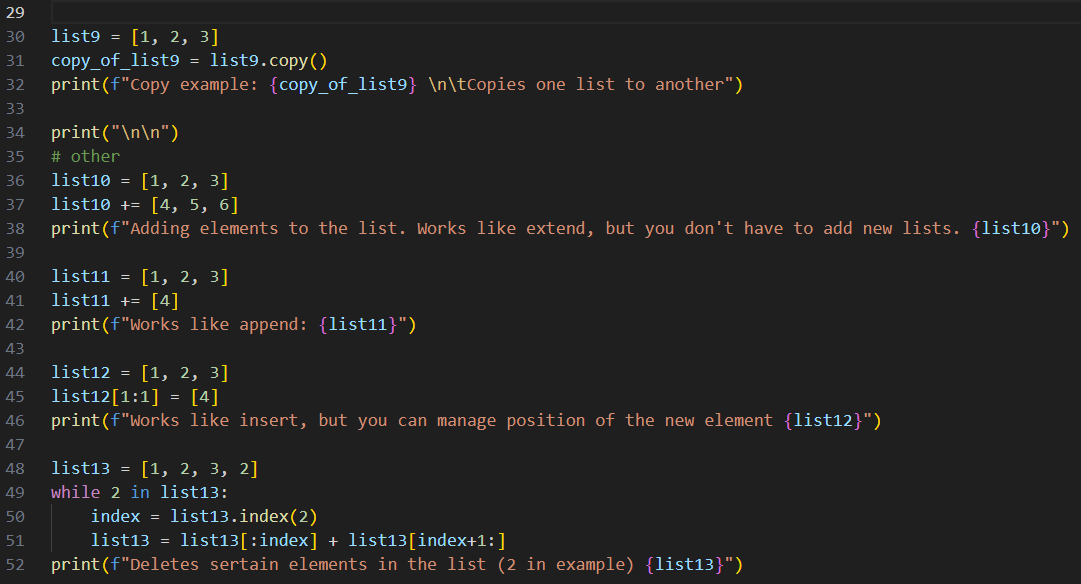
Перевірити поведінку функцій для списків: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy(), та інші.

Хід виконання завдання:

1. Знайшов інформацію по цим функціям.
2. Реалізував їх у коді.
3. Описав їх своїми словами, на англійській мові.

Текст програми:

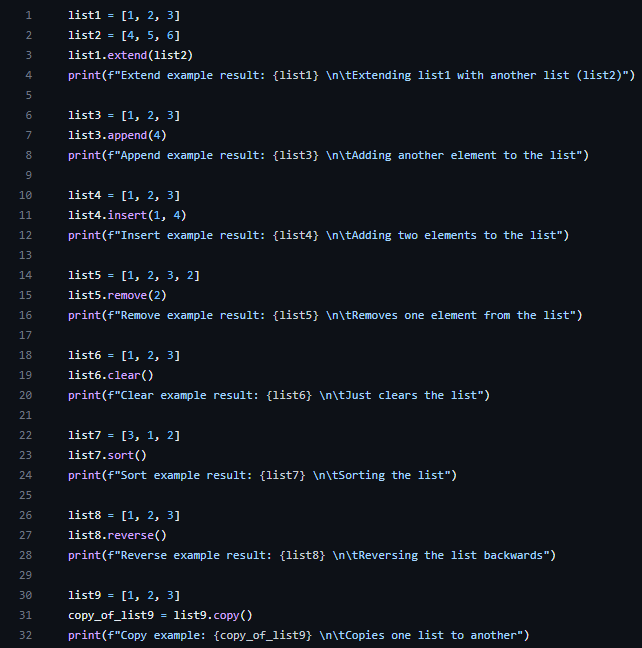


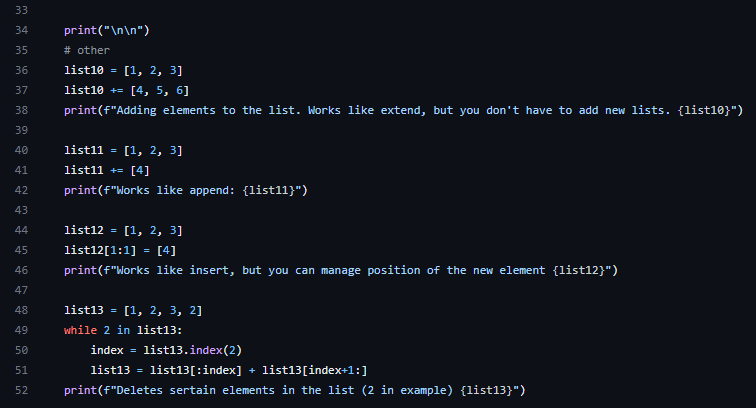


Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_03/task2.py)

Знімок екрану з посилання на github:





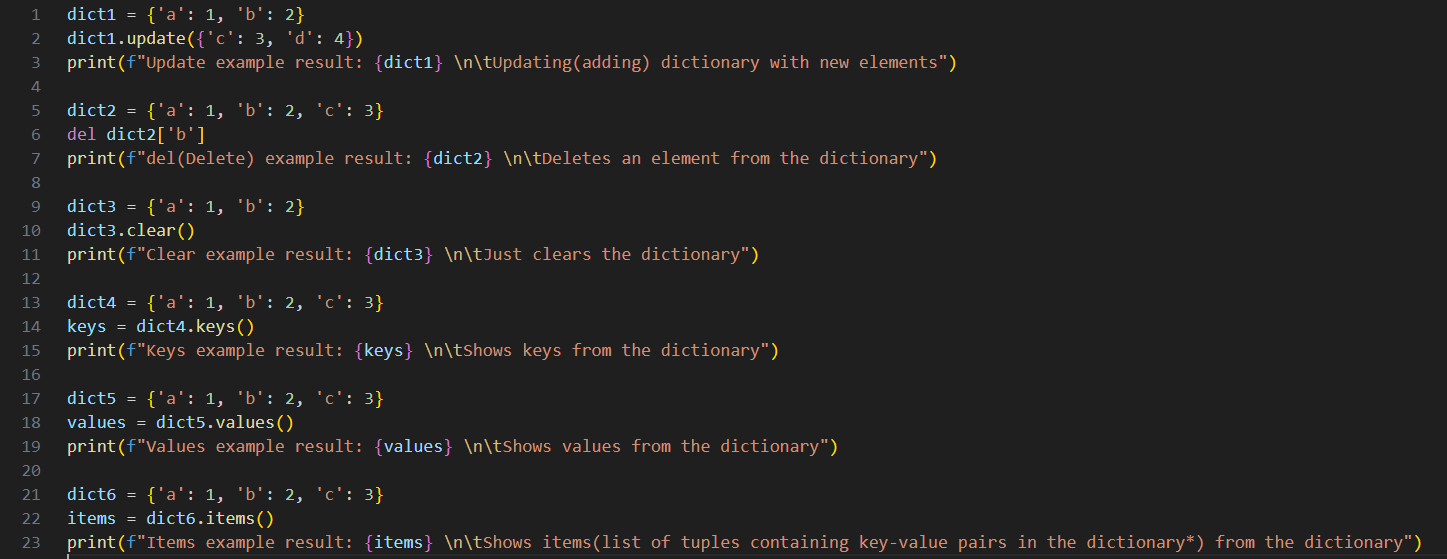
Завдання №3.3

Перевірити поведінку функцій для словників: update(), del, clear, keys(), values(), items().

Хід виконання завдання:

1. Знайшов інформацію по цим функціям.
2. Реалізував їх у коді.
3. Описав їх своїми словами, на англійській мові.

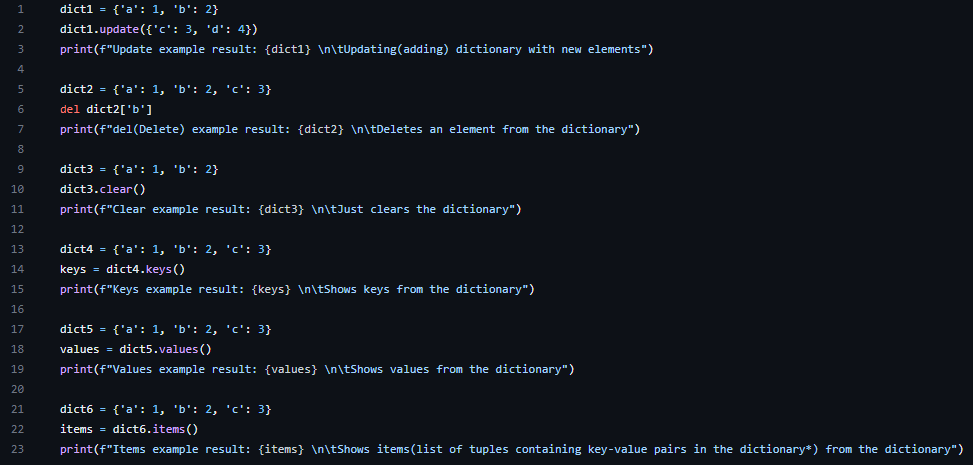
Текст програми:



Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_03/task3.py)

Знімок екрану з посилання на github:



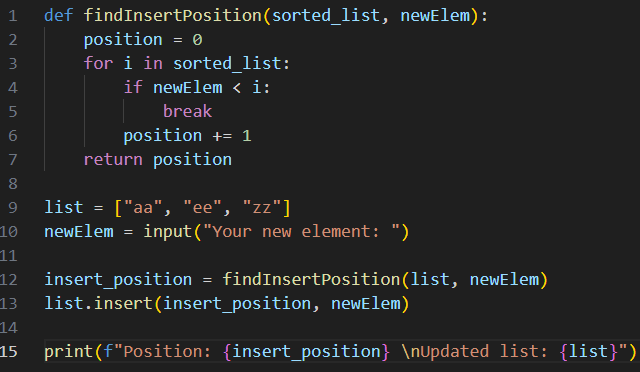
Завдання №3.4

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

Хід виконання завдання:

1. Уважно перечитав завдання, щоб добре осмислити задачу.
2. Написав функцію для пошуку позиції вставлення нового елементу в список, використовуючи цикл та умовний оператор.
3. Задав початковий та відсортований список, створив запит для нового елементу, який необхідно вставити.
4. Підставив аргументи взяті вище.
5. Через функцію insert() вставив новий елемент на позицію визначеною написаною функцією вище.
6. Вивів результати.

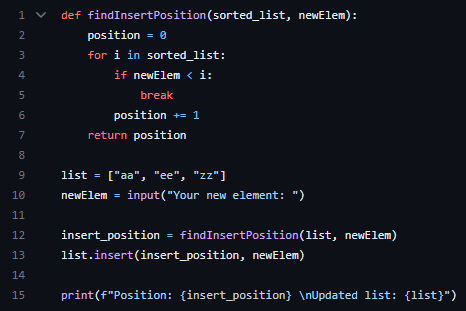
Текст програми:



Посилання на github:

[\*link\*](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_03/task4.py)

Знімок екрану з посилання на github:



Завдання №4.1-2

Розширити програму калькулятор функцією запитів від користувача, що обробляє виняткові ситуації.

Розширити функцію ділення обробкою виняткової ситуації ділення на нуль.

Хід виконання завдання:

1. Повторив матеріал з лекції.
2. Написав функцію getIntValue, з конструкцією try-except, метою якої є обробка виняткової ситуації, яка виникає коли користувач вводить неправильний тип даних.
3. Доробив функцію ділення, додавши до неї try-except, для того щоб уникнути помилки при випадку коли користувач намагається поділити число на нуль.

Текст програми:

def plus (a, b):

    return a + b

def minus (a, b):

    return a - b

def multiply (a, b):

    return a \* b

def divide (a, b):

    try:

        return a / b

    except ZeroDivisionError:

        print("You can't divide a number by zero. Try againg...")

def getIntValue(promt: str):

    while True:

        try:

            return int(input(promt))

        except ValueError:

            print("Value is not integer. Try againg...")

while True:

    operation = input("If you want to quit the program - type \"Quit\" \nWhat's your operation (+, -, \*, /): ")

    if operation == "Quit" or operation == "quit":

        print("Quiting the program...")

        break

    elif operation == "+" or operation == "-" or operation == "\*" or  operation == "\*" or operation == "/":

        a = getIntValue("Your first num: ")

        b = getIntValue("Your second num: ")

        match(operation):

            case "+":

                print("Answer is:", plus(a, b))

            case "-":

                print("Answer is:", minus(a, b))

            case "\*":

                print("Answer is:", multiply(a, b))

            case "/":

                print("Answer is:", divide(a, b))

            case \_:

                print("You did something wrong, try againg.")

    else:

        print("You did something wrong, write your operation (+, -, \*, /)")

Посилання на github:

[GitHub](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_04/task_01_02.py)

Знімок екрану з посилання на github:



Завдання №4.3

Ознайомитись зі списком виняткових ситуацій за посиланням <https://docs.python.org/3/library/exceptions.html>

Хід виконання завдання:

1. Ознайомився з матеріалом по посиланню.
2. Знайшов та подивився декілька відео з прикладами на цю тему.
3. Написав декілька конструкцій try-except.

Текст програми:

# ValueError

while True:

    try:

        x = int(input("Enter Int: "))

        print(x)

        break

    except ValueError:

        print("Your value isn't Integer. Print Int...")

# ZeroDivisionError, else, ValueError and Finally(works in any scenario)

while True:

    try:

        x = 5

        y = int(input("Divide 5 with any Int except 0: "))

        print(x/y)

    except ZeroDivisionError:

        print("You must type any Int except 0. Try again..")

    except ValueError:

        print("Your value isn't Integer. Print Int...")

    else:

        break

    finally:

        print("0\_0")

# KeyError

dict = {'x': 1, 'y': 2}

try:

    value = dict['z']

except KeyError:

    print("Key error. Key for 'value' doesn't exist")

# TypeError

try:

    result = "10" + 5

except TypeError as e:

    print("You can't add string to integer")

Посилання на github:

[GitHub](https://github.com/7878seveneight/TP-KB-222-Petro-Ratozii/blob/main/topic_04/task_03.py)

Знімок екрану з посилання на github:

