

# **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:

Я вирішив трохи ускладнити собі задачу, і зробив можливість вводу будь якого тексту, але з умовою, що цей текст буде мати 10 символів, якщо ні, то спрацюють обробники помилки і скажуть більше треба, чи менше, якщо ж текст має 10 символів, то за допомогою слайсів, де можна вибрати з якого елемента масива почати, де зупинитись і який крок, в моєму випадку я зазначив лише який крок -1, щоб він йшов з кінця.

Текст програми:

```
text = input("Введіть текст з 10 знаками: ")
```

```
if len(text) > 10:
```

```
print("Знаків більше ніж 10")
```

```
elif len(text) < 10:
```

```
print("Знаків менше ніж 10")
```

```
else:
```

```
reserved = text[::-1]
```

```
print(reserved)
```

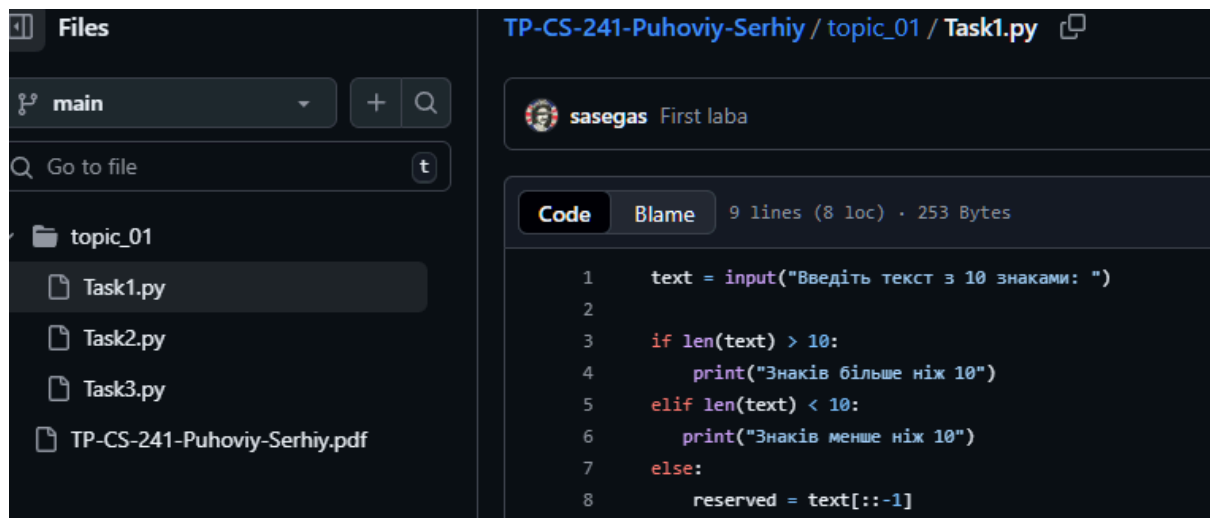


Рис.1 Скріншот першого завдання з GitHub

Посилання: [https://github.com/sasegas/TP-CS-241-Puhoviy-Serhiy/blob/main/topic\\_01/Task1.py](https://github.com/sasegas/TP-CS-241-Puhoviy-Serhiy/blob/main/topic_01/Task1.py)

## Стилі для тексту

Виконати деякі тести на strip, capitalize, title, upper, lower

Хід виконання завдання:

В цьому завданні я написав текст в змінній string, щоб його можна було перевірити на всі вищезазначені функції,

strip - прибирає зайві пробіли на початку і в кінці,

capitalize - в рядку тільки перша літера, першого слова велика,

title - в кожному слові великі літери, тільки перші, а інші в нижньому регістрі,

upper - всі букви у верхньому регістрі,

lower - всі букви у нижньому регістрі

Текст програми:

```
string = '    hello pyTHon!    '
print(string)
```

```
print('strip:',string.strip())
print('capitalize:',string.strip().capitalize())
print('title:',string.strip().title())
print('upper:',string.strip().upper())
print('lower:',string.strip().lower())
```

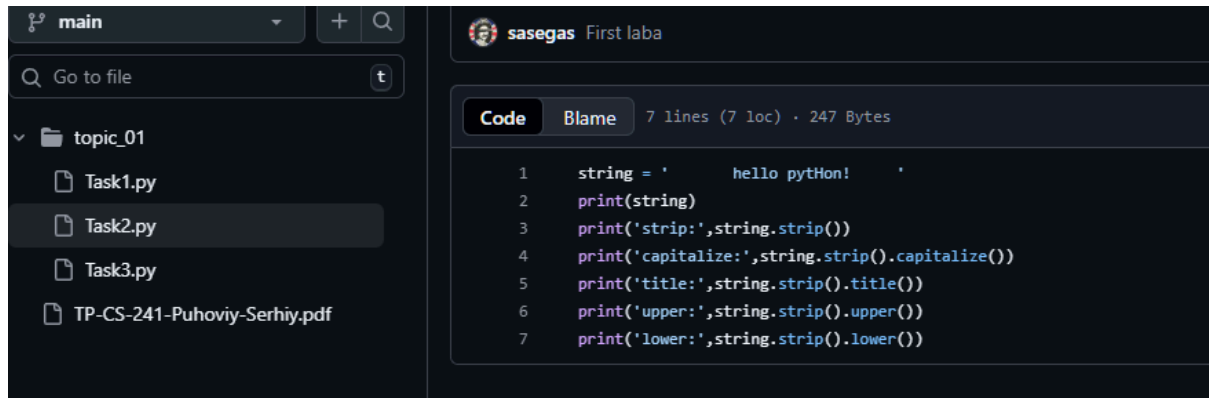


Рис.2 Скріншот другого завдання з GitHub

Посилання:[https://github.com/sasegas/TP-CS-241-Puhoviy-Serhiy/blob/main/topic\\_01/Task2.py](https://github.com/sasegas/TP-CS-241-Puhoviy-Serhiy/blob/main/topic_01/Task2.py)

## Квадратне рівняння

Знайти відповідь для квадратного рівняння за допомогою функцій

Хід виконання завдання:

Спочатку я створив функцію `quadraticFn` в якій задав 3 змінні,  $a, b, c$  відповідно як у рівнянні і після цього задав змінну дискримінанта, де його обрахував відповідно до формули і після цього використав умовний оператор `if`, для того щоб правильно порахувати  $x$ , після чого ми маємо три варіанти відповіді, або  $x$  не існує, або є тільки один  $x$ , або є  $x_1$  і  $x_2$ . Для того щоб порахувати квадратний корінь дискримінанту мені знадобилась бібліотека `math`.

## Текст програми:

```
import math

def quadraticFn():

    a = int(input('Введіть перший коефіцієнт:'))
    b = int(input('Введіть другий коефіцієнт:'))
    c = int(input('Введіть вільний член:'))

    discriminator = b**2 - 4*a*c

    if discriminator == 0:

        print ("Рівняння має один корінь")

        x = -b/(2*a)

        print(x)

    elif discriminator > 0:

        print ("Рівняння має два корінь")

        x1 = (-b - math.sqrt(discriminator))/(2*a)
        x2 = (-b + math.sqrt(discriminator))/(2*a)

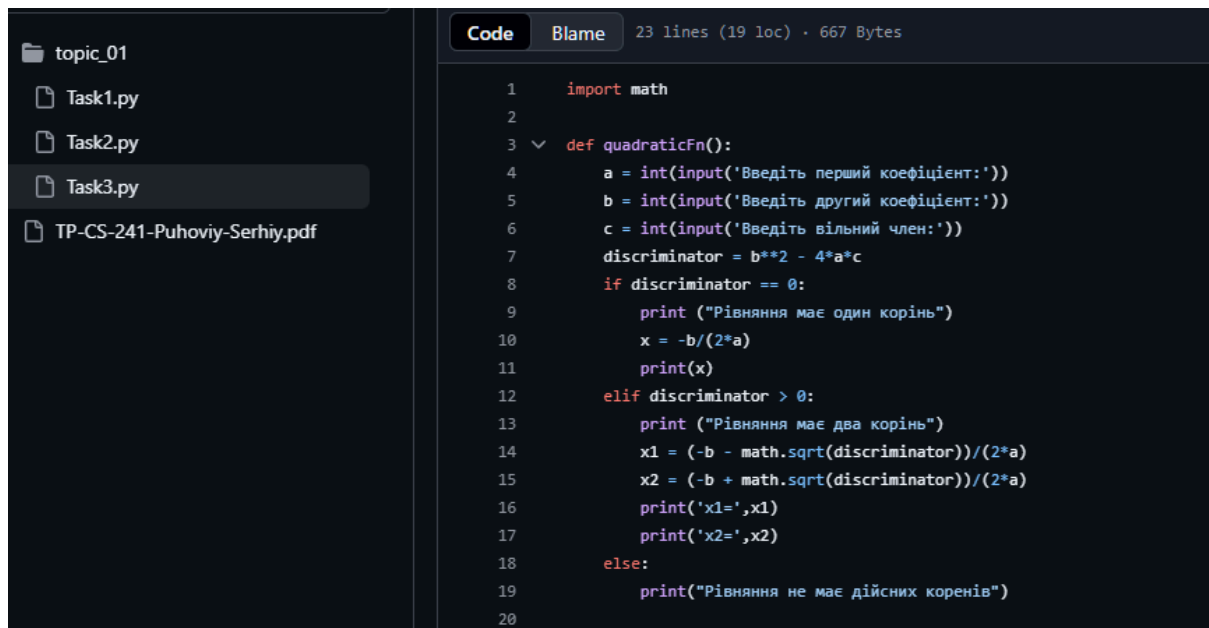
        print('x1=', x1)

        print('x2=', x2)

    else:

        print("Рівняння не має дійсних коренів")

quadraticFn()
```



```
1 import math
2
3 def quadraticFn():
4     a = int(input('Введіть перший коефіцієнт:'))
5     b = int(input('Введіть другий коефіцієнт:'))
6     c = int(input('Введіть вільний член:'))
7     discriminator = b**2 - 4*a*c
8     if discriminator == 0:
9         print("Рівняння має один корінь")
10        x = -b/(2*a)
11        print(x)
12    elif discriminator > 0:
13        print("Рівняння має два коріння")
14        x1 = (-b - math.sqrt(discriminator))/(2*a)
15        x2 = (-b + math.sqrt(discriminator))/(2*a)
16        print('x1=', x1)
17        print('x2=', x2)
18    else:
19        print("Рівняння не має дійсних коренів")
20
```

Рис.3 Скріншот третього завдання з GitHub

Посилання: [https://github.com/sasegas/TP-CS-241-Puhoviy-Serhiy/blob/main/topic\\_01/Task2.py](https://github.com/sasegas/TP-CS-241-Puhoviy-Serhiy/blob/main/topic_01/Task2.py)

## Звіт до Теми №2

### Умовний перехід

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

#### Функція пошуку коренів

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

Хід виконання завдання:

Використав код для знаходження дискримінанта з попередньої теми, який я додав, після нього я написав функцію, для розрахунку дискримінанта, за допомогою умовних операторів if, elif, else, я додав три умови по яким і знаходяться корені.

Текст програми:

```
import math

def discriminant(a,b,c):

    return b**2 - 4*a*c


def quadratic_roots(a, b, c):

    d = discriminant(a, b, c)

    if d > 0:

        print("Рівняння має два корені")

        x1 = (-b - math.sqrt(d)) / (2*a)

        x2 = (-b + math.sqrt(d)) / (2*a)

        print("x1 =", x1)

        print("x2 =", x2)

        return x1, x2

    elif d == 0:

        print("Рівняння має один корінь")

        x = -b / (2*a)

        print("x =", x)

        return x,

    else:

        print("Рівняння не має дійсних коренів")

        return None


a, b, c = 3, -18, 27

roots = quadratic_roots(a, b, c)
```

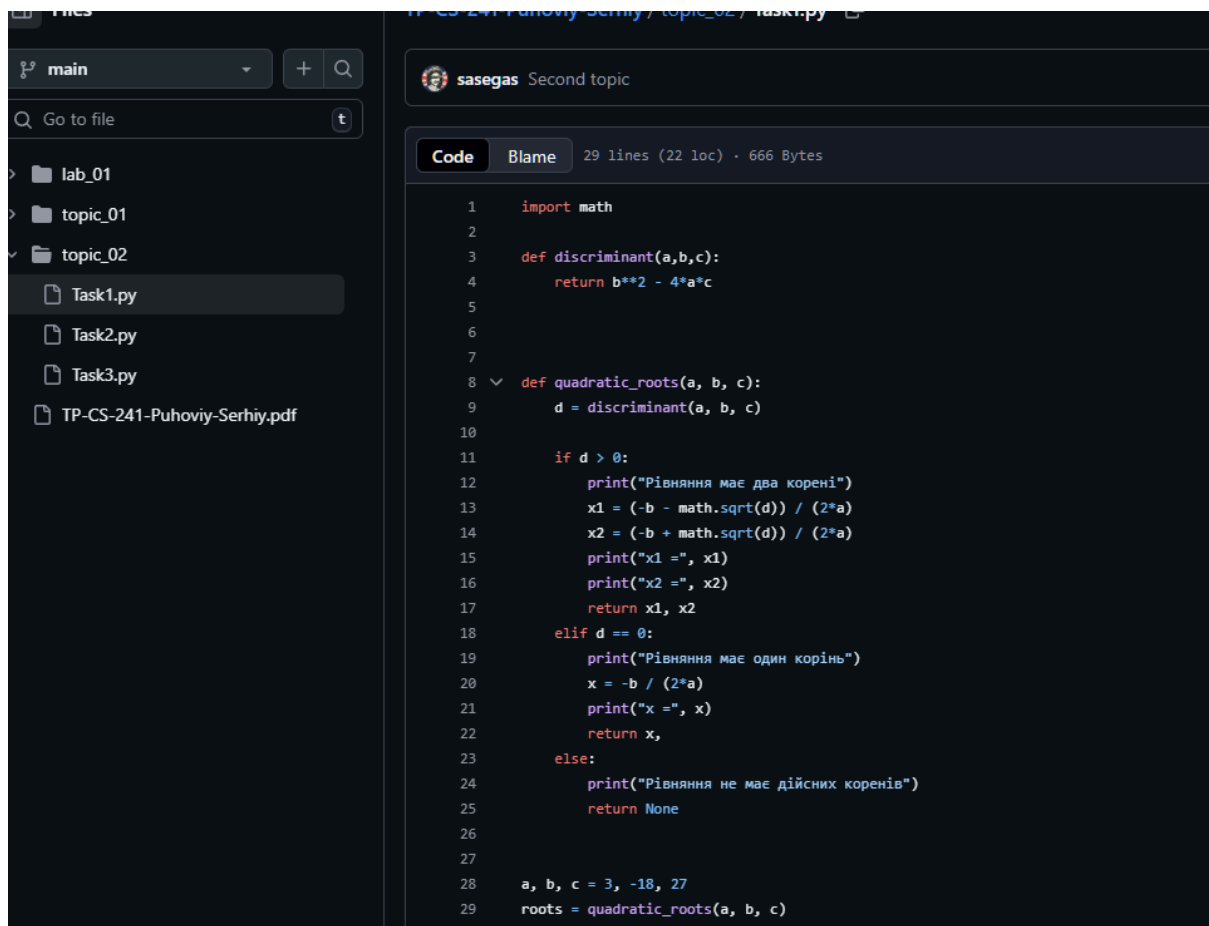


Рис.1 Скріншот першого завдання з GitHub

## Калькулятор використовуючи if

Написати програму калькулятор використовуючи if else конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

В цьому завданні я спочатку додав input для кожного значення, а саме першого числа, другого і знаку, потім створив функцію калькулятор, який за допомогою умовних операторів робить певну дію, чи то множення, ділення, чи інші, а потім значення цієї функції поміщається в змінну result, яку, потім показую за допомогою print

Текст програми:

```
a = float(input("введіть перше число: "))
b = float(input("введіть перше друге число: "))
sign = input("введіть дію(+ - * /): ")
```

```
def calculator (a, b, sign):  
    if sign == "+":  
        return a + b  
    elif sign == "-":  
        return a - b  
    elif sign == "*":  
        return a * b  
    elif sign == "/":  
        if b != 0:  
            return a / b  
        else:  
            return "Ділити на 0 неможна"  
    else:  
        return "Була задана некоректна дія"  
  
result = calculator(a,b,sign)  
print("Відповідь: ", result)
```



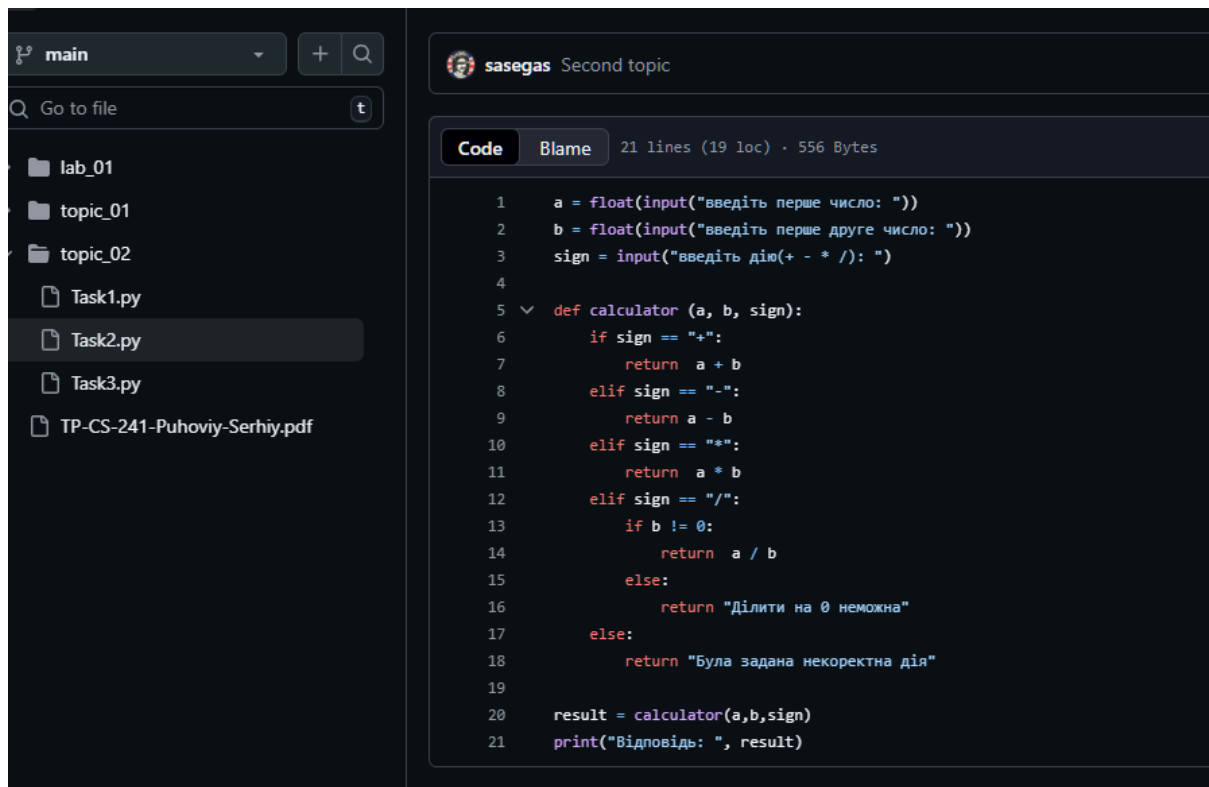


Рис.2 Скріншот другого завдання з GitHub

### Калькулятор використовуючи match

Написати програму калькулятор використовуючи match конструкцію. Кожна операція має бути виконана в окремій функції.

Хід виконання завдання:

В цьому завданні я зробив 4 окремі функції, які відповідають за конкретну дію і за допомогою match запускав певну дію в залежності від вибору користувача, а потім виводив

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):  
    if b!= 0:  
        return a/b  
    else:  
        return "Ділити на 0 неможна"
```

```
a = float(input("введіть перше число: "))
b = float(input("введіть перше друге число: "))
sign = input("введіть дію(+ - * /): ")

match(sign):
    case "+":
        result = plus(a,b)

    case "-":
        result = minus(a,b)

    case "*":
        result = multiply(a,b)

    case "/":
        result = divide(a,b)
    case _:
        result = "Була задана некоректна дія"

print("Відповідь: ", result)
```

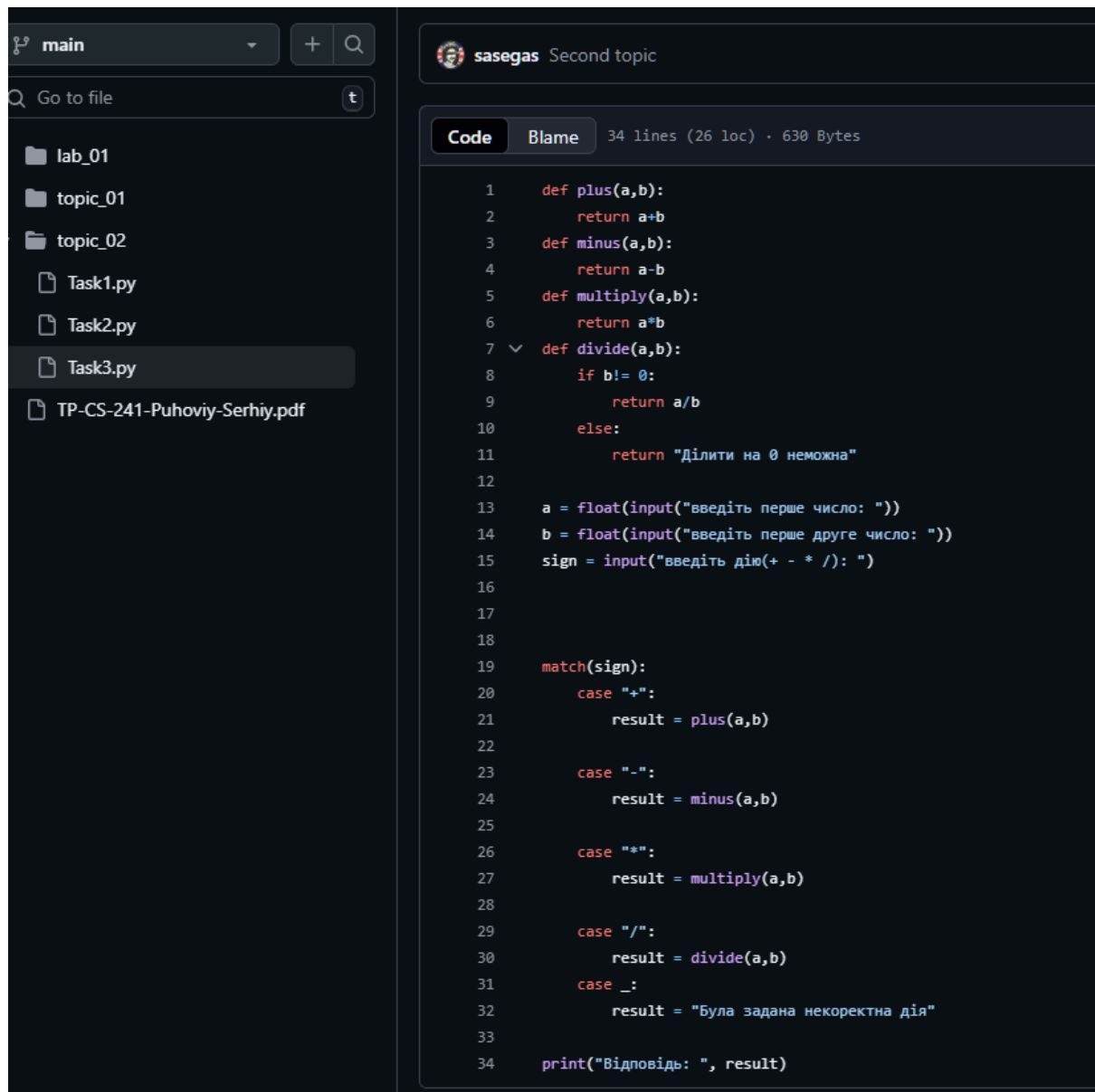


Рис.3 Скріншот третього завдання з GitHub