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

Райчинець Олеся КБ-242

Звіт до Теми №1

Функції та змінні

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

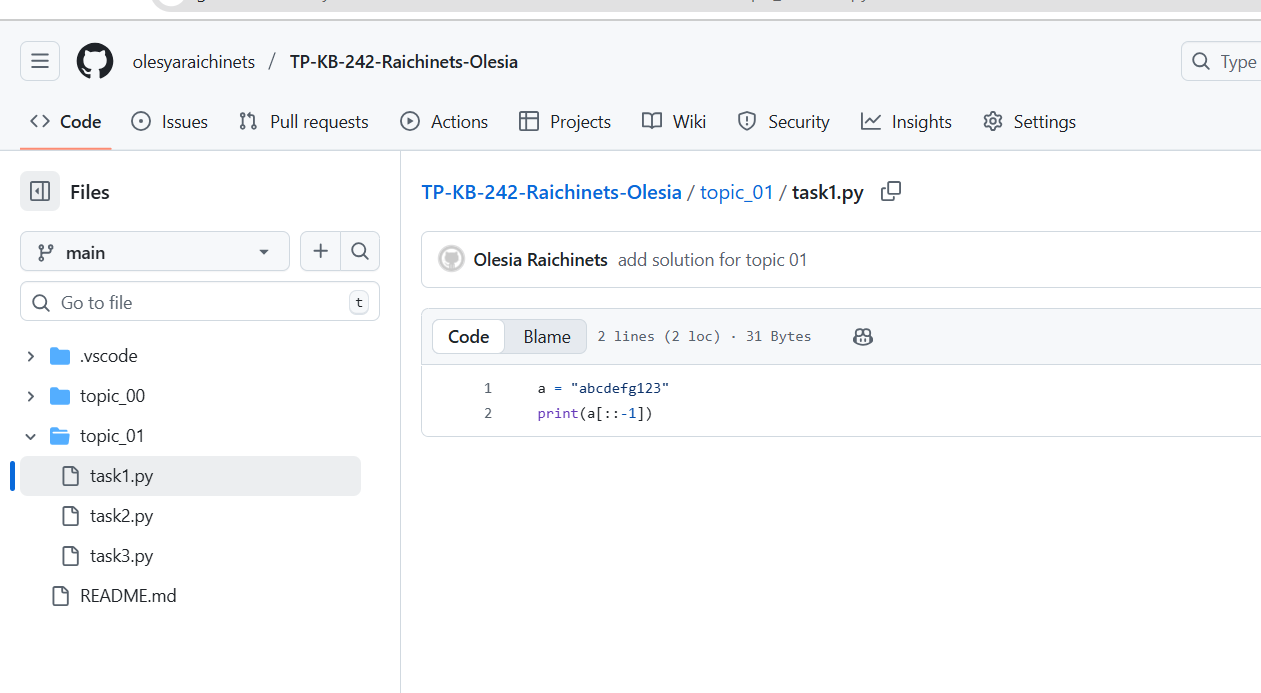
**1.Перетворення рядка**

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

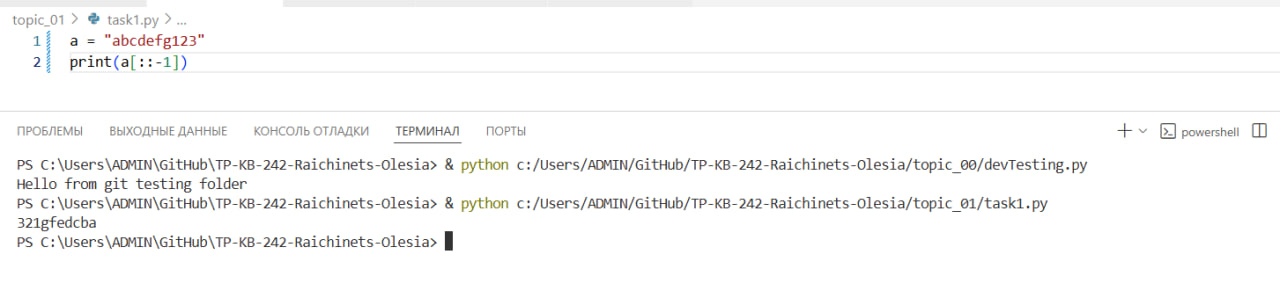
Текст програми:

|  |
| --- |
| a = "abcdefg123"  print(a[::-1]) |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_01/task1.py>

Знімок екрану з посилання на github: 

Знімок результату:

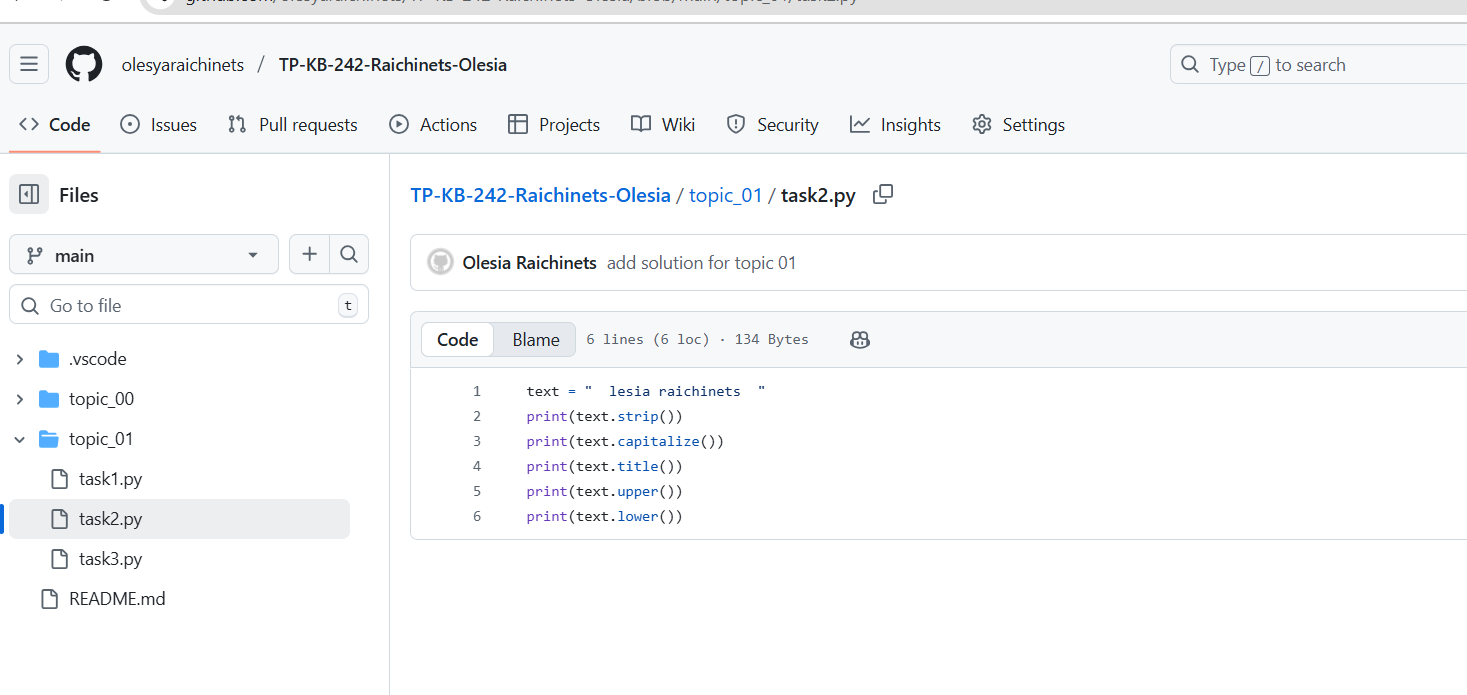


**2.Виконати деякі тести для strip, capitalize, title, upper, lower**

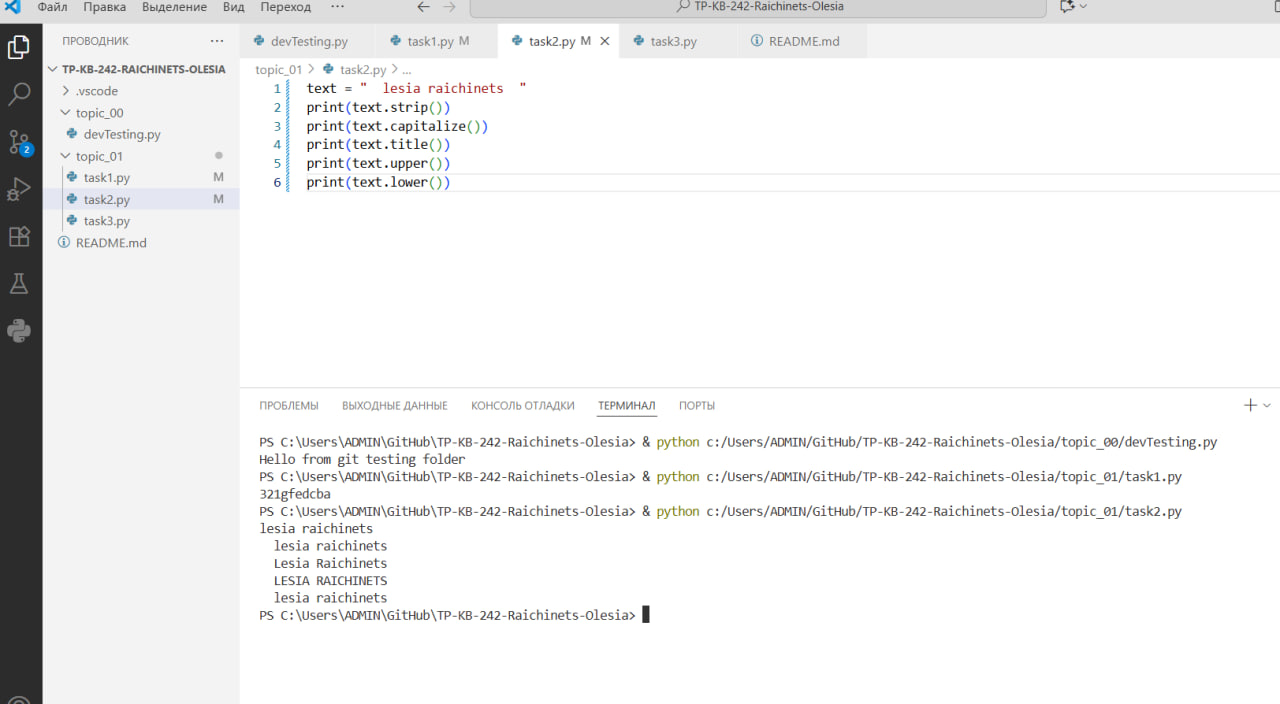
Текст програми:

|  |
| --- |
| text = " lesia raichinets "  print(text.strip())  print(text.capitalize())  print(text.title())  print(text.upper())  print(text.lower()) |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_01/task2.py>

Знімок екрану з посилання на github: 

Знімок результату:



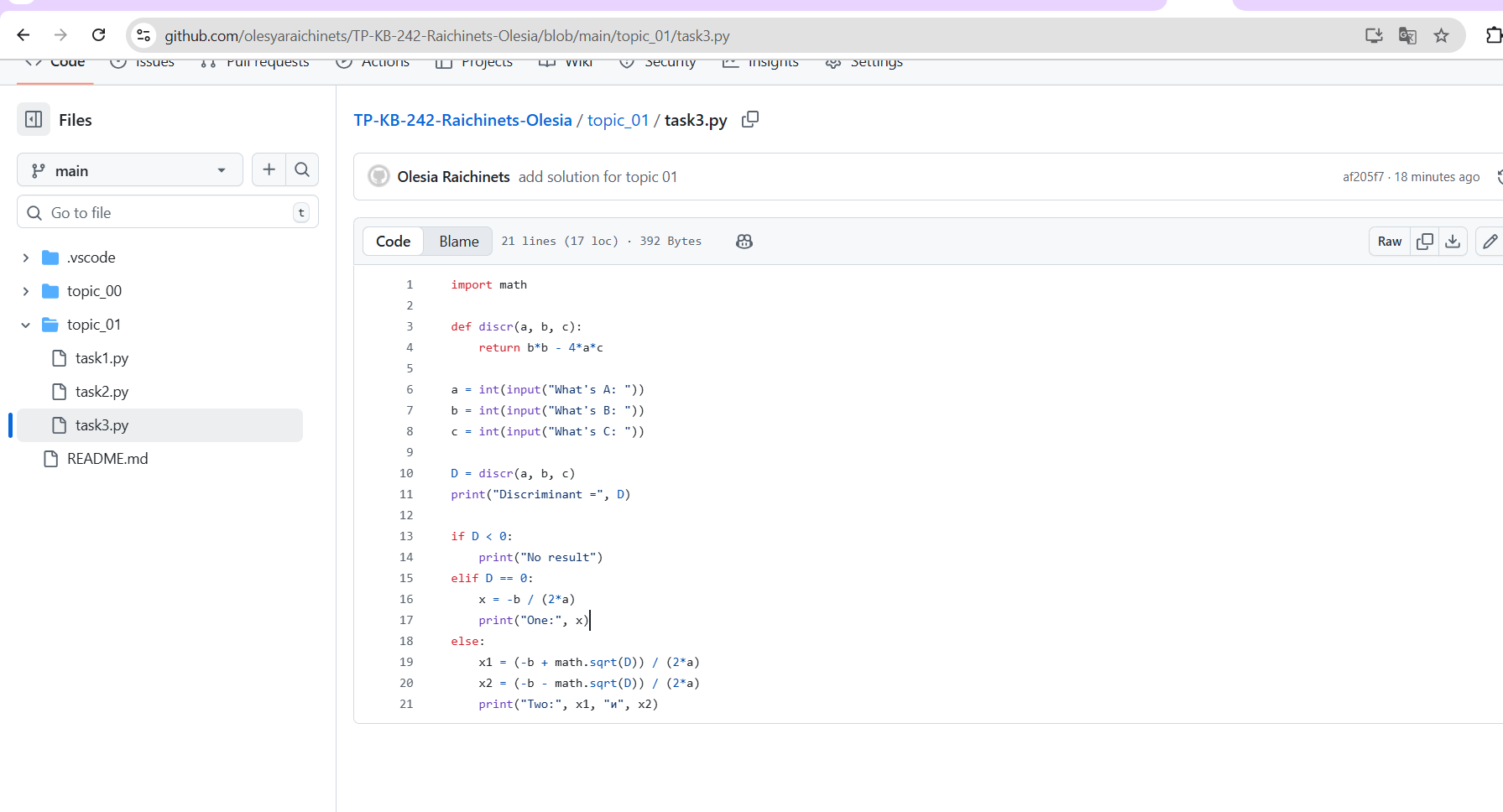
**3.Дискримінант**

Необхідно знайти відповідь для квадратного рівняння за допомогою функцій

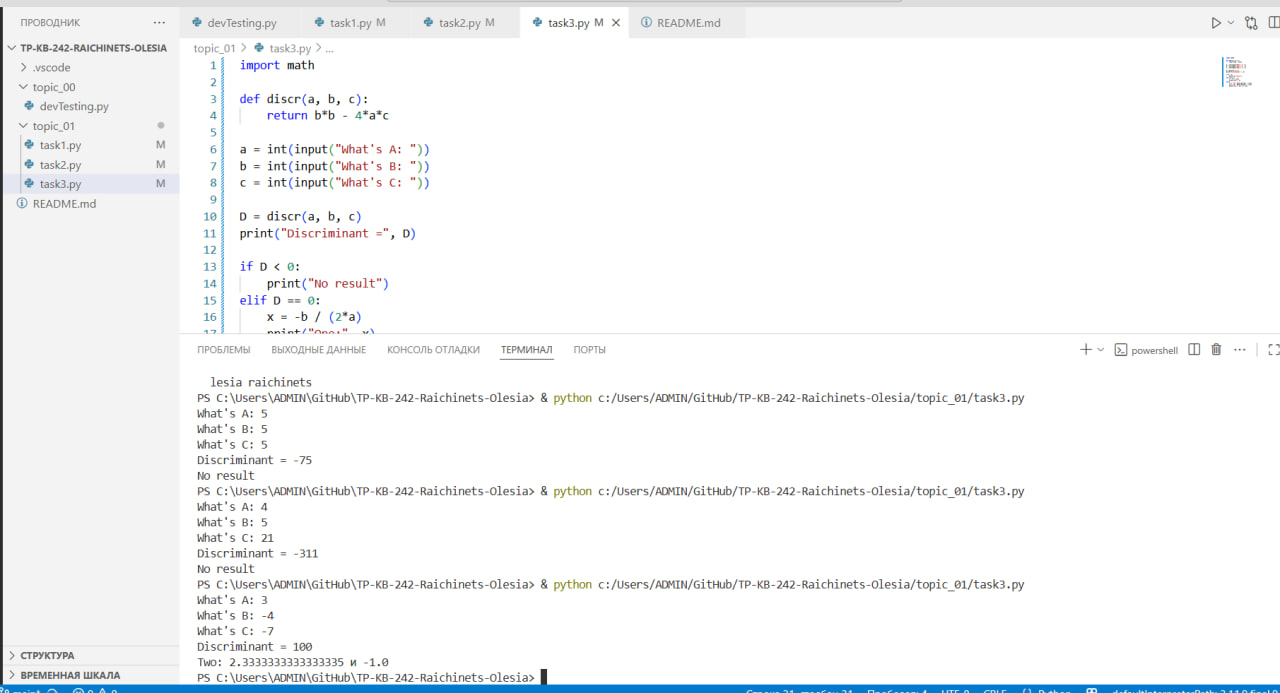
Текст програми:

|  |
| --- |
| import math  def discr(a, b, c):  return b\*b - 4\*a\*c  a = int(input("What's A: "))  b = int(input("What's B: "))  c = int(input("What's C: "))  D = discr(a, b, c)  print("Discriminant =", D)  if D < 0:  print("No result")  elif D == 0:  x = -b / (2\*a)  print("One:", x)  else:  x1 = (-b + math.sqrt(D)) / (2\*a)  x2 = (-b - math.sqrt(D)) / (2\*a)  print("Two:", x1, "и", x2) |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_01/task3.py>

Знімок екрану з посилання на github: 

Знімок результату:



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

Райчинець Олеся КБ-242

Звіт до Теми 2

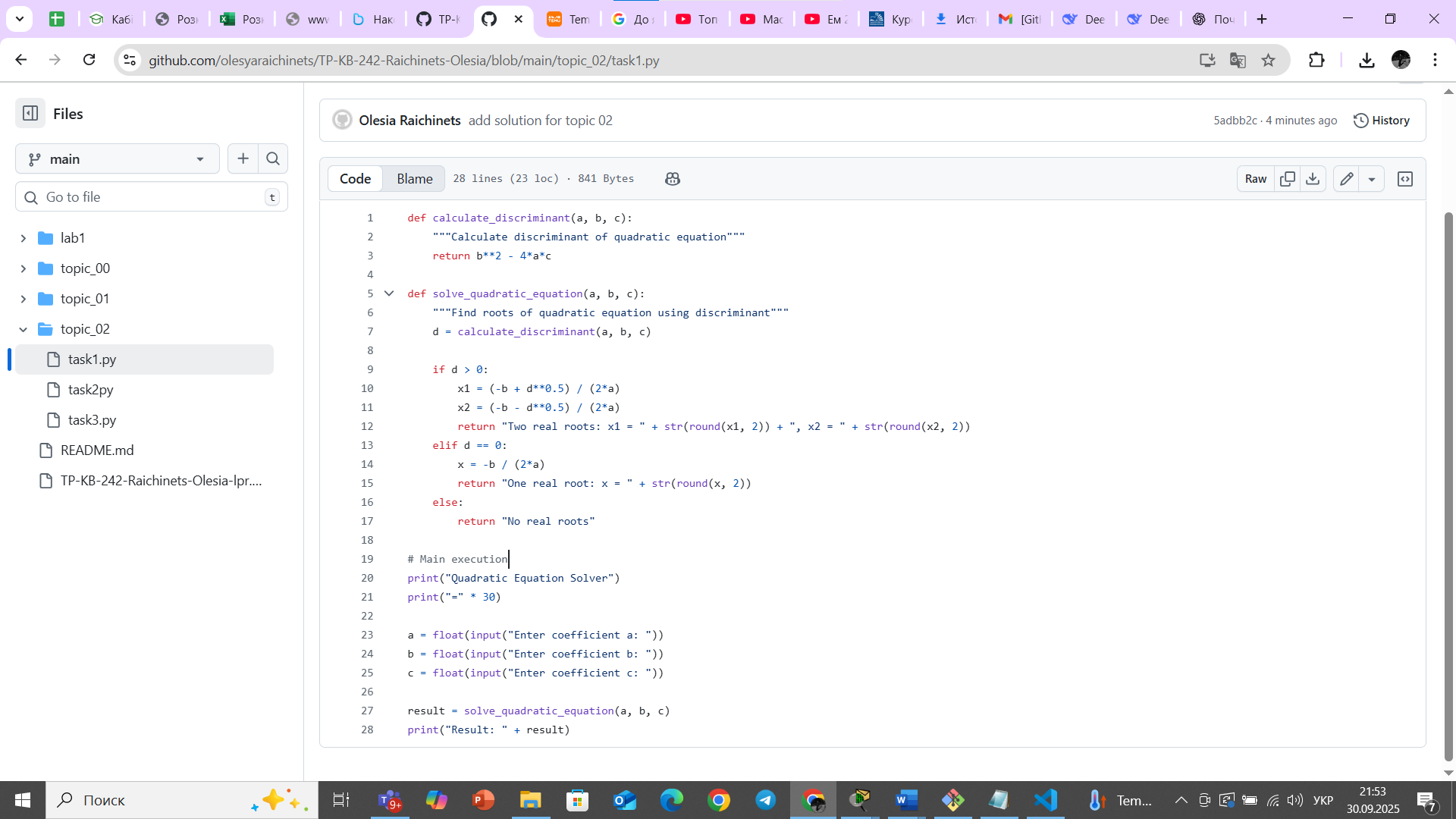
1. Написати функцію пошуку коренів квадратного рівняння використовуючи функцію розрахунку дискримінанту з попередньої теми та умовні переходи.

Текст програми:

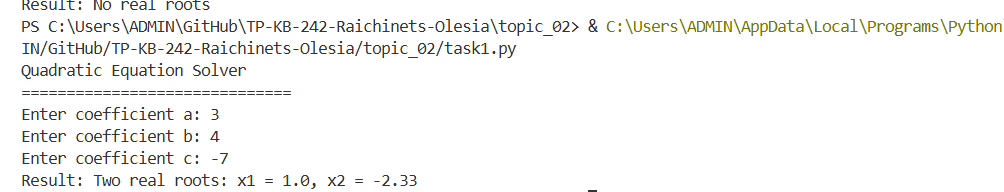
|  |
| --- |
| def calculate\_discriminant(a, b, c):      """Calculate discriminant of quadratic equation"""      return b\*\*2 - 4\*a\*c  def solve\_quadratic\_equation(a, b, c):      """Find roots of quadratic equation using discriminant"""      d = calculate\_discriminant(a, b, c)        if d > 0:          x1 = (-b + d\*\*0.5) / (2\*a)          x2 = (-b - d\*\*0.5) / (2\*a)          return "Two real roots: x1 = " + str(round(x1, 2)) + ", x2 = " + str(round(x2, 2))      elif d == 0:          x = -b / (2\*a)          return "One real root: x = " + str(round(x, 2))      else:          return "No real roots"  # Main execution  print("Quadratic Equation Solver")  print("=" \* 30)  a = float(input("Enter coefficient a: "))  b = float(input("Enter coefficient b: "))  c = float(input("Enter coefficient c: "))  result = solve\_quadratic\_equation(a, b, c)  print("Result: " + result) |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_02/task1.py>

Знімок екрану з посилання на github:



Знімок результату:



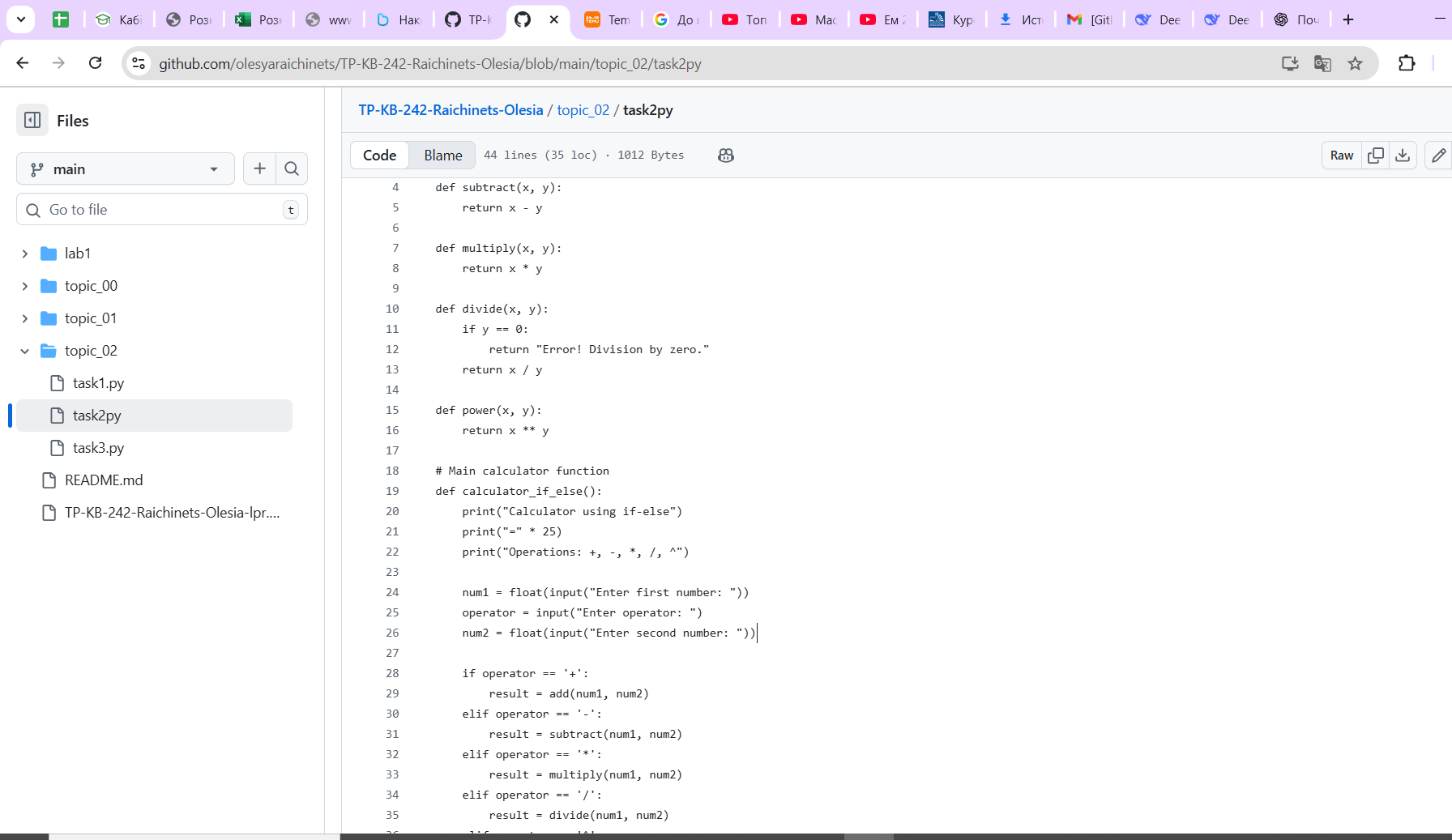
1. Написати програму калькулятор використовуючи **if else** конструкцію. Кожна операція має бути виконана в окремій функції.

Текст програми:

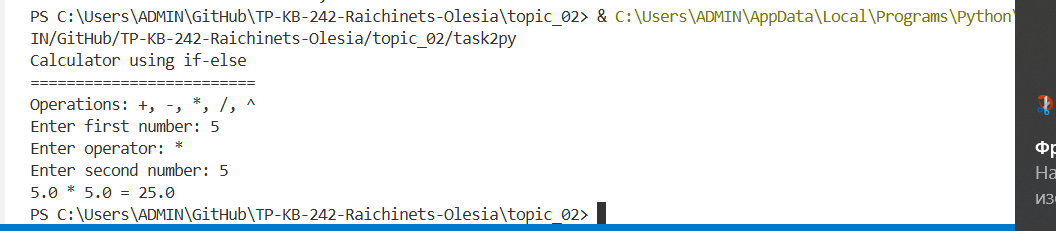
|  |
| --- |
| def add(x, y):  return x + y  def subtract(x, y):  return x - y  def multiply(x, y):  return x \* y  def divide(x, y):  if y == 0:  return "Error! Division by zero."  return x / y  def power(x, y):  return x \*\* y  # Main calculator function  def calculator\_if\_else():  print("Calculator using if-else")  print("=" \* 25)  print("Operations: +, -, \*, /, ^")    num1 = float(input("Enter first number: "))  operator = input("Enter operator: ")  num2 = float(input("Enter second number: "))    if operator == '+':  result = add(num1, num2)  elif operator == '-':  result = subtract(num1, num2)  elif operator == '\*':  result = multiply(num1, num2)  elif operator == '/':  result = divide(num1, num2)  elif operator == '^':  result = power(num1, num2)  else:  result = "Invalid operator!"    print(str(num1) + " " + operator + " " + str(num2) + " = " + str(result))  # Run calculator  calculator\_if\_else() |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_02/task2py>

Знімок екрану з посилання на github:



Знімок результату:

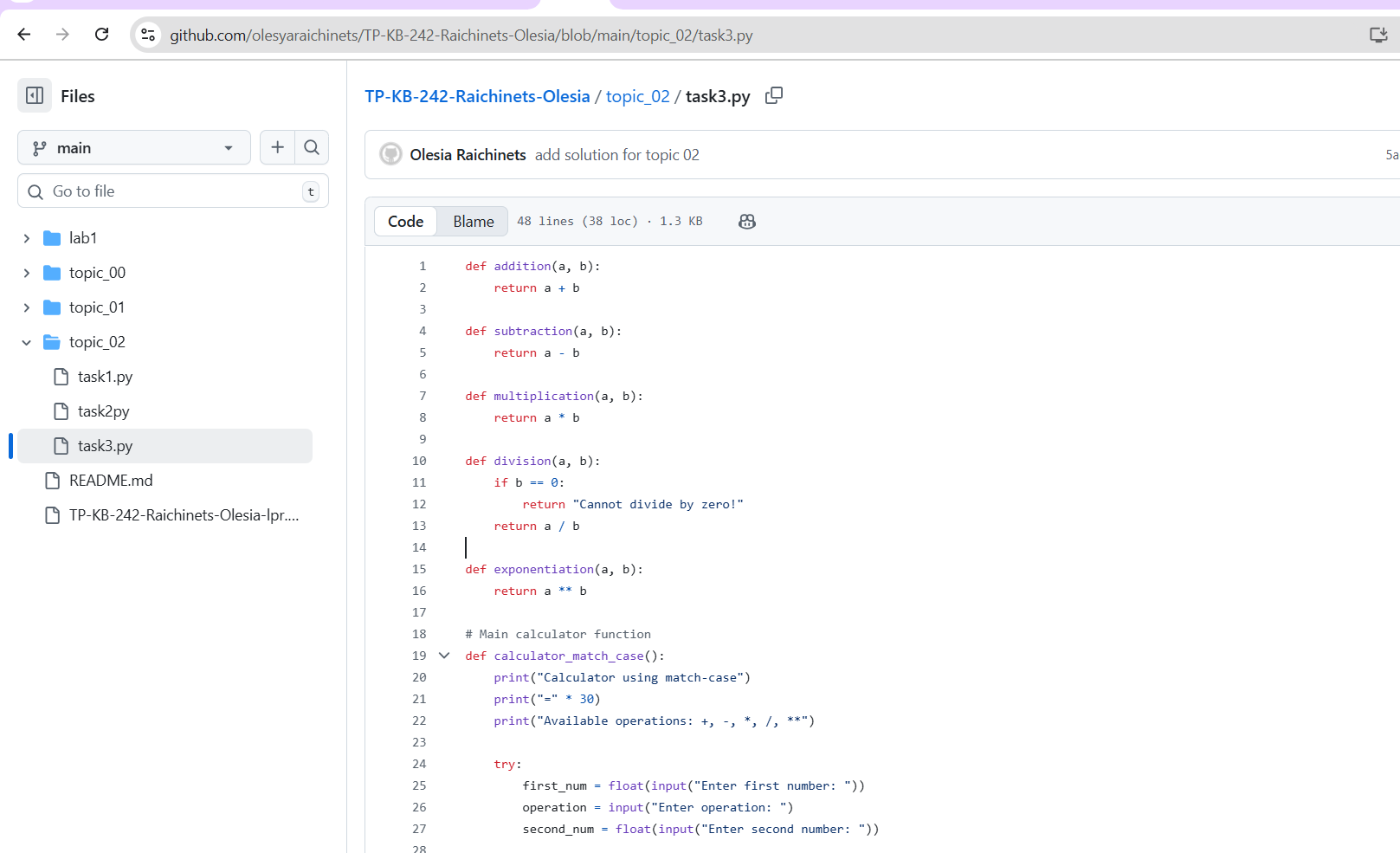


1. Написати програму калькулятор використовуючи **match** конструкцію. Кожна операція має бути виконана в окремій функції.

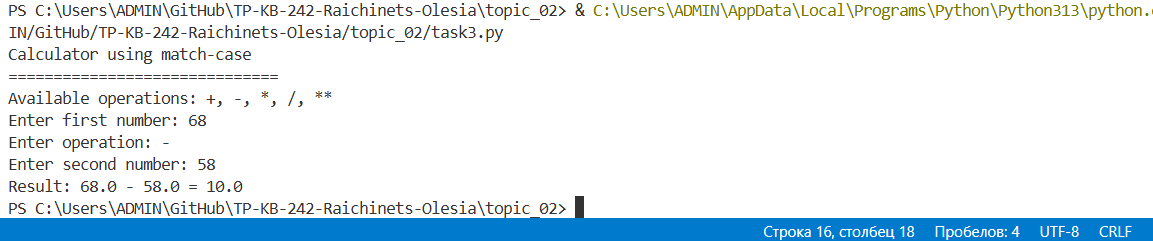
Текст програми:

|  |
| --- |
| def addition(a, b):      return a + b  def subtraction(a, b):      return a - b  def multiplication(a, b):      return a \* b  def division(a, b):      if b == 0:          return "Cannot divide by zero!"      return a / b  def exponentiation(a, b):      return a \*\* b  # Main calculator function  def calculator\_match\_case():      print("Calculator using match-case")      print("=" \* 30)      print("Available operations: +, -, \*, /, \*\*")        try:          first\_num = float(input("Enter first number: "))          operation = input("Enter operation: ")          second\_num = float(input("Enter second number: "))            match operation:              case '+':                  result = addition(first\_num, second\_num)              case '-':                  result = subtraction(first\_num, second\_num)              case '\*':                  result = multiplication(first\_num, second\_num)              case '/':                  result = division(first\_num, second\_num)              case '\*\*':                  result = exponentiation(first\_num, second\_num)              case \_:                  result = "Unknown operation!"            print("Result: " + str(first\_num) + " " + operation + " " + str(second\_num) + " = " + str(result))        except ValueError:          print("Please enter valid numbers!")  calculator\_match\_case() |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_02/task3.py>

Знімок екрану з посилання на github: 

Знімок результату:



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

Райчинець Олеся КБ-242

Звіт до Теми 3

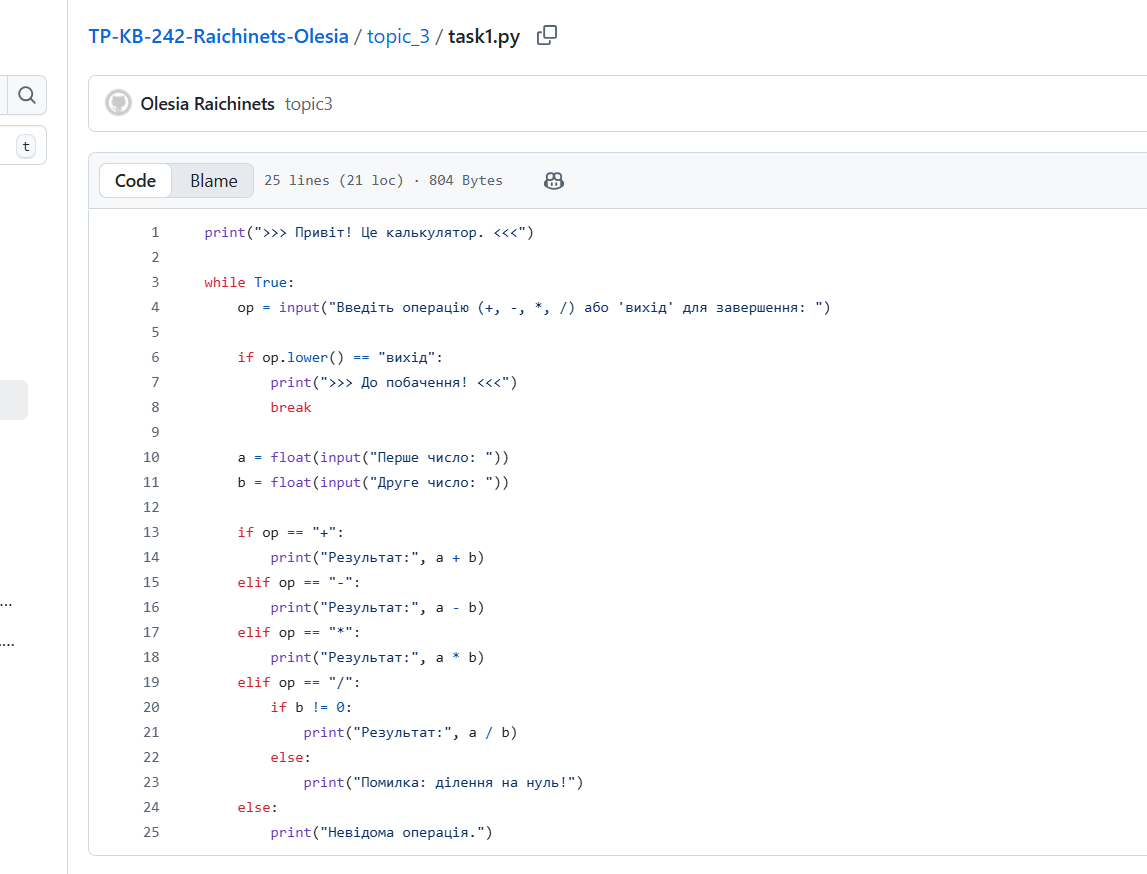
1. Написати програму калькулятор з постійними запитами на введення нових даних та операцій. За основу взяти програму калькулятор з попередньої теми. Реалізувати механізм завершення програми після отримання відповідної команди.

Текст програми:

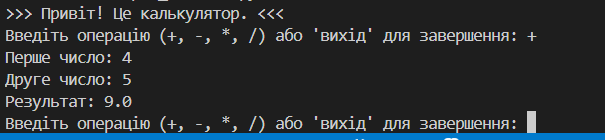
|  |
| --- |
| print(">>> Привіт! Це калькулятор. <<<")  while True:      op = input("Введіть операцію (+, -, \*, /) або 'вихід' для завершення: ")      if op.lower() == "вихід":          print(">>> До побачення! <<<")          break      a = float(input("Перше число: "))      b = float(input("Друге число: "))      if op == "+":          print("Результат:", a + b)      elif op == "-":          print("Результат:", a - b)      elif op == "\*":          print("Результат:", a \* b)      elif op == "/":          if b != 0:              print("Результат:", a / b)          else:              print("Помилка: ділення на нуль!")      else:          print("Невідома операція.") |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_3/task1.py>

Знімок екрану з посилання на github:



Знімок результату:



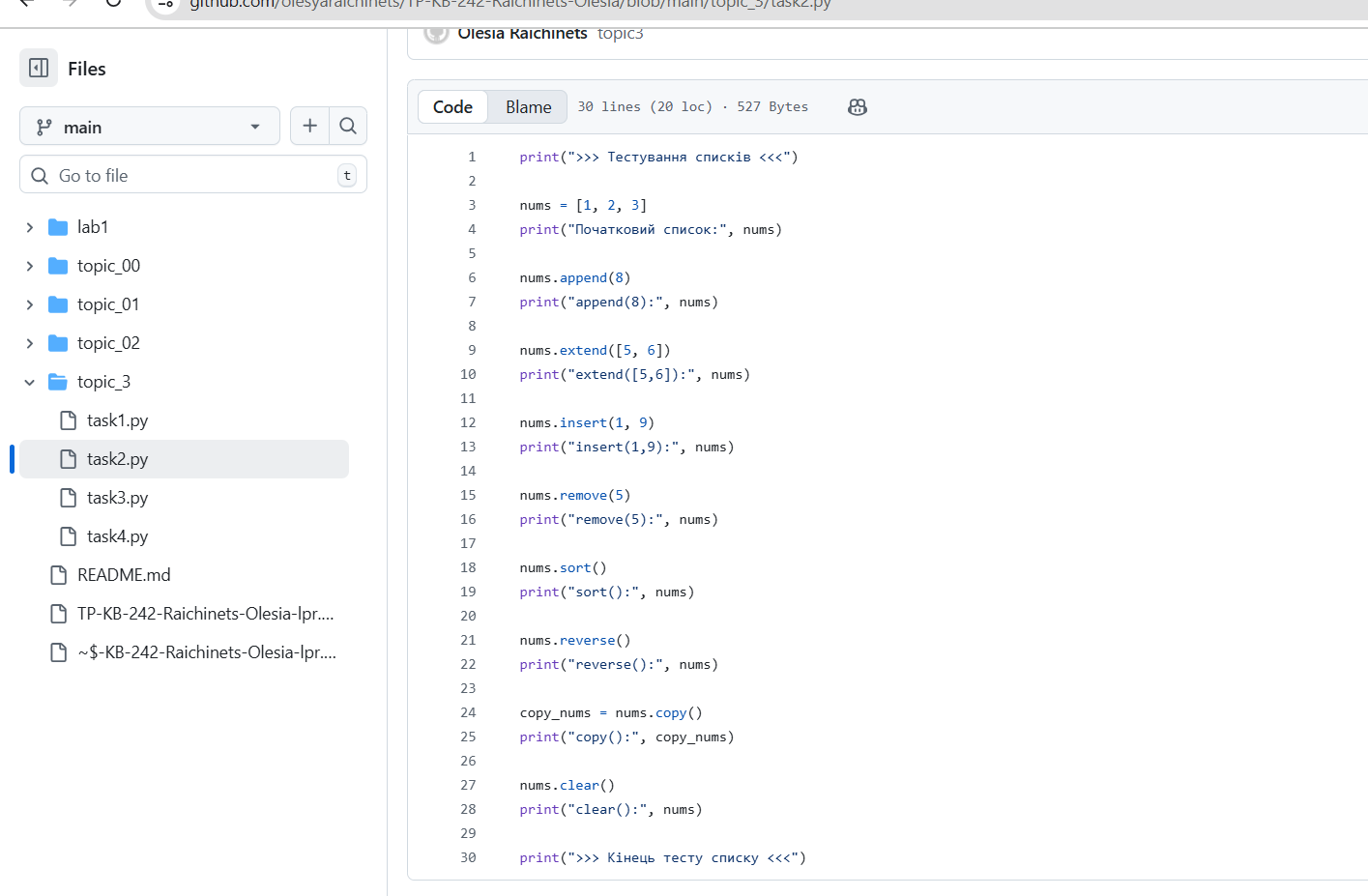
1. Написати програму тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy()

Текст програми:

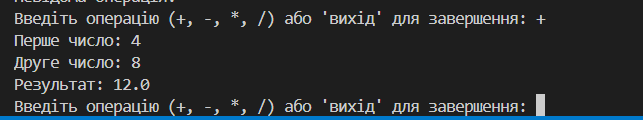
|  |
| --- |
| print(">>> Тестування списків <<<")  nums = [1, 2, 3]  print("Початковий список:", nums)  nums.append(8)  print("append(8):", nums)  nums.extend([5, 6])  print("extend([5,6]):", nums)  nums.insert(1, 9)  print("insert(1,9):", nums)  nums.remove(5)  print("remove(5):", nums)  nums.sort()  print("sort():", nums)  nums.reverse()  print("reverse():", nums)  copy\_nums = nums.copy()  print("copy():", copy\_nums)  nums.clear()  print("clear():", nums)  print(">>> Кінець тесту списку <<<") |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_3/task2.py>

Знімок екрану з посилання на github:



Знімок результату:



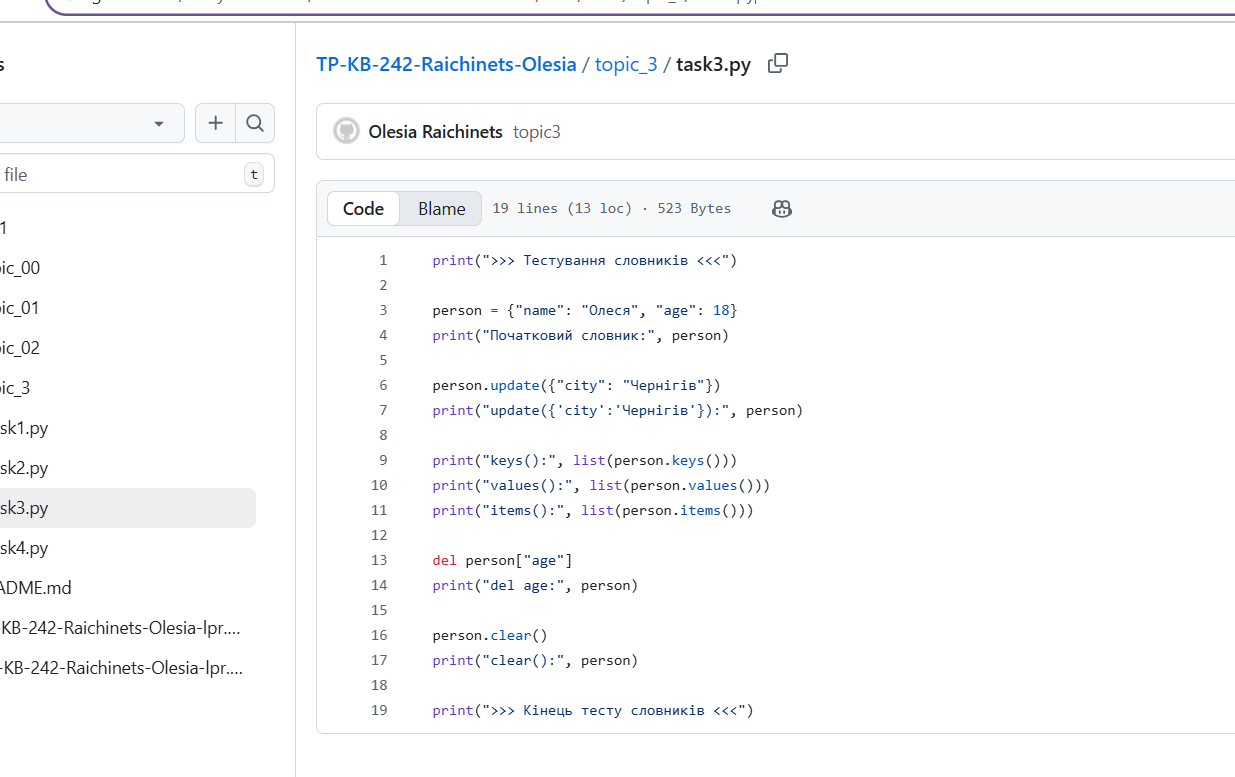
1. Написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items()

Текст програми:

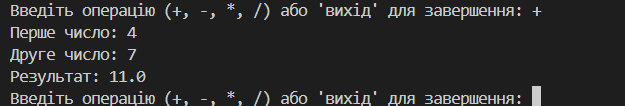
|  |
| --- |
| print(">>> Тестування словників <<<")  person = {"name": "Олеся", "age": 18}  print("Початковий словник:", person)  person.update({"city": "Чернігів"})  print("update({'city':'Чернігів'}):", person)  print("keys():", list(person.keys()))  print("values():", list(person.values()))  print("items():", list(person.items()))  del person["age"]  print("del age:", person)  person.clear()  print("clear():", person)  print(">>> Кінець тесту словників <<<") |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_3/task3.py>

Знімок екрану з посилання на github:



Знімок результату:



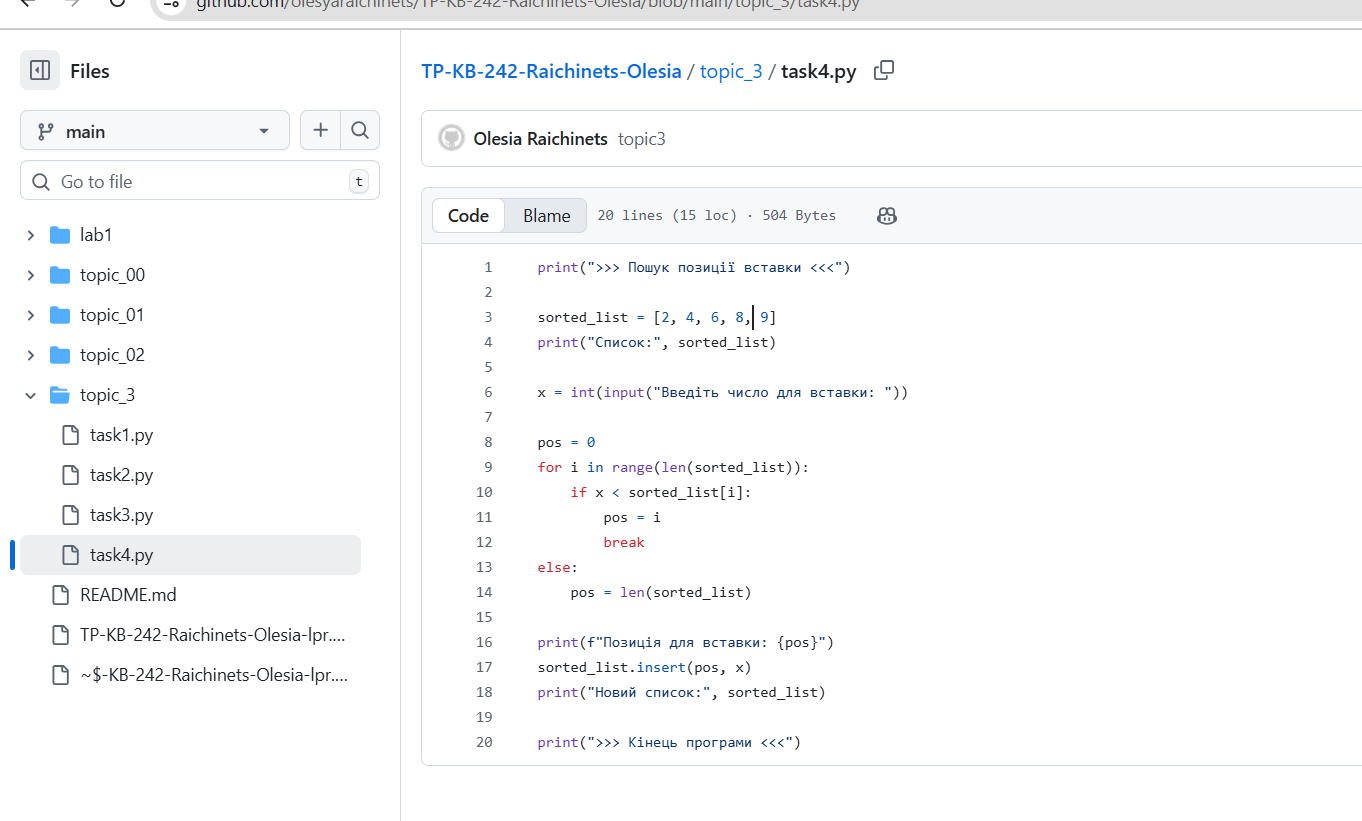
1. Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

Текст програми:

|  |
| --- |
| print(">>> Пошук позиції вставки <<<")  sorted\_list = [2, 4, 6, 8, 9]  print("Список:", sorted\_list)  x = int(input("Введіть число для вставки: "))  pos = 0  for i in range(len(sorted\_list)):      if x < sorted\_list[i]:          pos = i          break  else:      pos = len(sorted\_list)  print(f"Позиція для вставки: {pos}")  sorted\_list.insert(pos, x)  print("Новий список:", sorted\_list)  print(">>> Кінець програми <<<") |

Посилання на github: <https://github.com/olesyaraichinets/TP-KB-242-Raichinets-Olesia/blob/main/topic_3/task4.py>

Знімок екрану з посилання на github:



Знімок результату:

