**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:

1. Рядок, що має вигляд "abcdefg123" перетворити наступним чином "321gfedcba.

2. Тестування таких властивостей як: strip, capitalize, title, upper, lower.

3. Виконати пошук дискримінанту.

**Хід виконання завдання:**

**1.** Задав змінній значення (abcdefg123) та допомогою команди str[::-1] отримав зворотній рядок та вивів результат на екран.

**2.1**  Задав змінній значення (.lock.) та за допомогою функції strip(".") видалив всі крапки, які знаходяться на початку і в кінці рядка.

**2.2** Задав змінній значення (lock, local) та за допомогою функції capitalize() вивів перше слово з великої букви на екран.

**2.3** Задав змінній значення (lock, local) та за допомогою функції title() вивів всі слова з великої букви.

**2.4** Задав змінній значення (lock) та за допомогою функції upper() вивів слово великими буквами.

**2.5** Задав змінній значення (LOCK) та за допомогою функції lower() вивів слово маленькими літерами.

**3.** Задаю назву функції. Функція дискримінант з трьома параметрами також прописую формулу для D.

Повертаю значення D за допомогою команди return та задаю значення параметрам через консоль.

Потім повертаю назад D яке дорівнює дискримінанту. Result присвоюю discriminant та вивожу.

**Текст програми:**

**1.** str = ("abcdefg123")

print (str[::-1])

**2.** str = ".lock." # Видалення символів '.' з обох кінців рядка.

str = str.strip(".")

print(str)

str = "lock, local"

str = str.capitalize() # Перше слово з великої літери

print(str)

str = "lock, local"

str = str.title() # Всі слова з великої літери

print(str)

str = "lock"

str = str.upper() # Всі слова великими літерами

print(str)

str = "LOCK"

str = str.lower() # Всі слова маленькими літерами

print(str)

**3.**

def discriminant(a,b,c):

D = b\*b-4\*a\*c

return D

a = int(input("Please enter coef a : "))

b = int(input("Please enter coef b : "))

c = int(input("Please enter coef c : "))

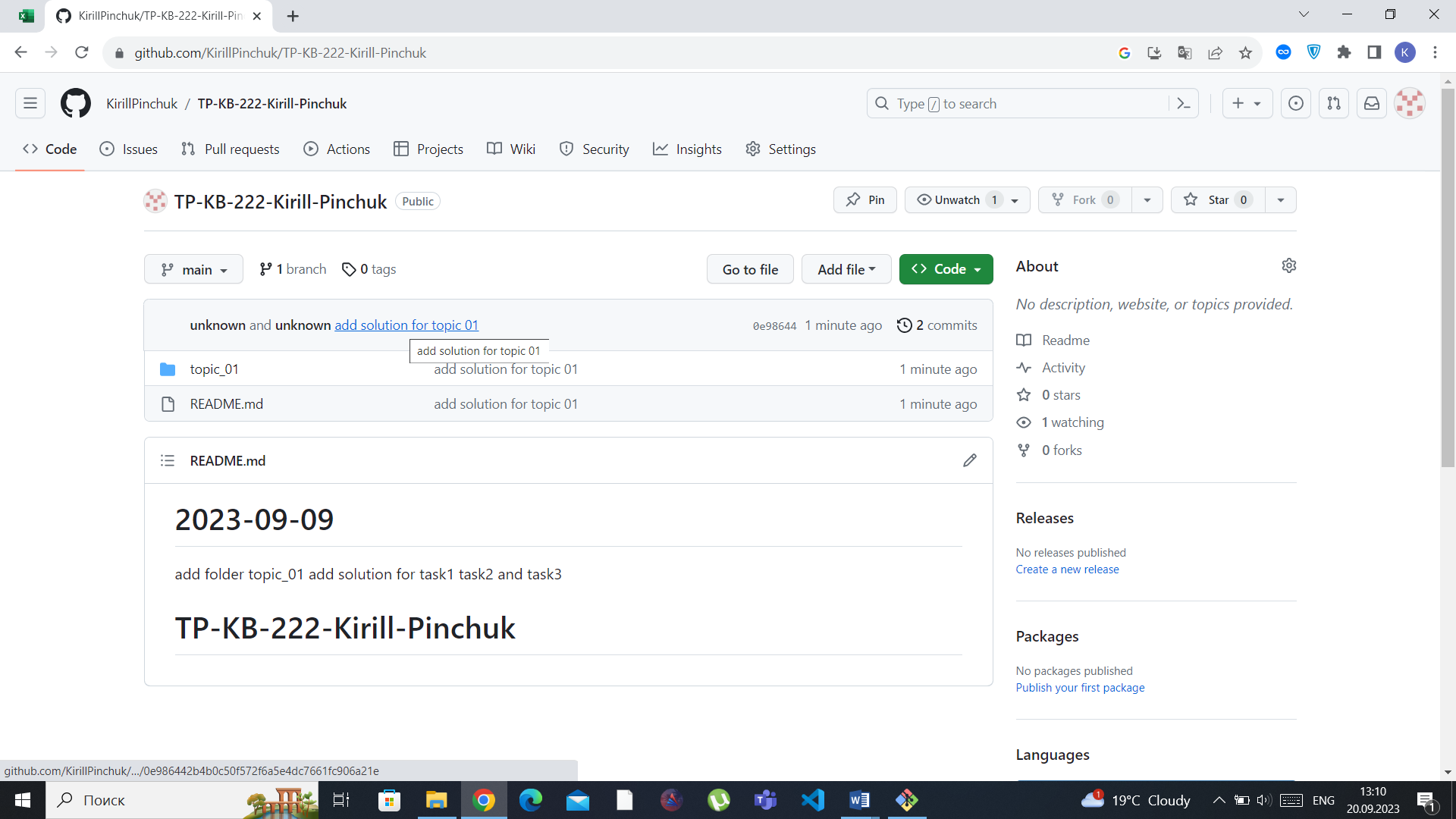
D = discriminant(a,b,c)

result = discriminant(a,b,c)

print(result)

**Посилання на github:** https://github.com/KirillPinchuk

**Знімок екрану з посилання на github:**



**Звіт до Теми №2**

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

1. Написати функцію для квадратного рівняння з використанням умови для D: D < 0, D == 0, D > 0.

2. Написати програму калькулятор з використанням if, elif.

3. Написати програму калькулятор з використаням match.

**Хід виконання завдання:**

1. Ввожу значення a, b, c за допомогою imput. Функція discr(a, b, c) обчислює дискримінант (D) за формулою D = b\*\*2 – 4\*a\*c та повертає його значення. Потім перевіряю D > 0, D<0 або D == 0 та вивожу результат від залежності значення дискримінанту. Також добавлена функція обчислення коренів при D > 0, D<0 або D == 0.
2. За допомогою **input**, **op** задаю значення a, b та операцію яку треба виконати. Додаю чотири функції для виконання математичних операцій: sum, minus, multiplication та division. В залежності від введеної операції, програма виконує відповідну функцію та зберігає результат в змінну result. Умовний оператор перевіряє введену операцію та виконує відповідну гілку коду. Якщо введена операція не відповідає жодній з чотирьох допустимих операцій ("+", "-", "\*", "/"), то програма виводить "Invalid operation".Результат обчислення або повідомлення про помилку виводяться на екран.
3. За допомогою **input**, **op** задаю значення a, b та операцію яку треба виконати.. Використовуючи match програма перевіряє значення op і визначає, яку операцію виконувати. У відповідності до значення op програма виконує одну з наступних дій: sum, minus, multiplication та division. Якщо op дорівнює "/", то програма перевіряє, чи b не дорівнює нулю. Якщо b не дорівнює нулю, то виконується функція division(a, b), і результат (частка a на b) виводиться на екран. В іншому випадку, якщо b дорівнює нулю, виводиться повідомлення "Invalid operation".

**Текст програми:**

**1**. a = int(input("Enter a: "))

b = int(input("Enter b: "))

c = int(input("Enter c: "))

def discr(a, b, c):

D = b\*\*2 - 4 \* a \* c

return D

D = discr(a, b, c)

if D > 0:

result = "D more 0"

print(D)

elif D == 0:

result = "D equal 0 "

else:

result = "No solution"

print(result)

**2.** a = int(input("Enter a: "))

b = int(input("Enter b: "))

op = input("Operation: ")

def sum(a, b):

return a + b

def minus(a,b):

return a - b

def multiplication(a,b):

return a \* b

def division(a, b):

if b != 0:

return a / b

else:

print("Division by zero is not allowed")

if op == "+":

result = sum(a,b)

print (result)

elif op == "-":

result = minus(a,b)

print (result)

elif op == "\*":

result = multiplication(a,b)

print (result)

elif op == "/":

result = division(a,b!=0)

print (result)

else:

print("Invalid operation")

**3.** a = int(input("Enter a: "))

b = int(input("Enter b: "))

op = (input("Operation: "))

match op:

case "+":

def sum(a, b):

return a + b

print (a + b)

case "-":

def minus(a,b):

return a - b

print (a -b)

case "\*":

def multiplication(a,b):

return a \* b

print (a \* b)

case "/":

if b != 0:

def division(a, b):

return a / b

result = division(a, b)

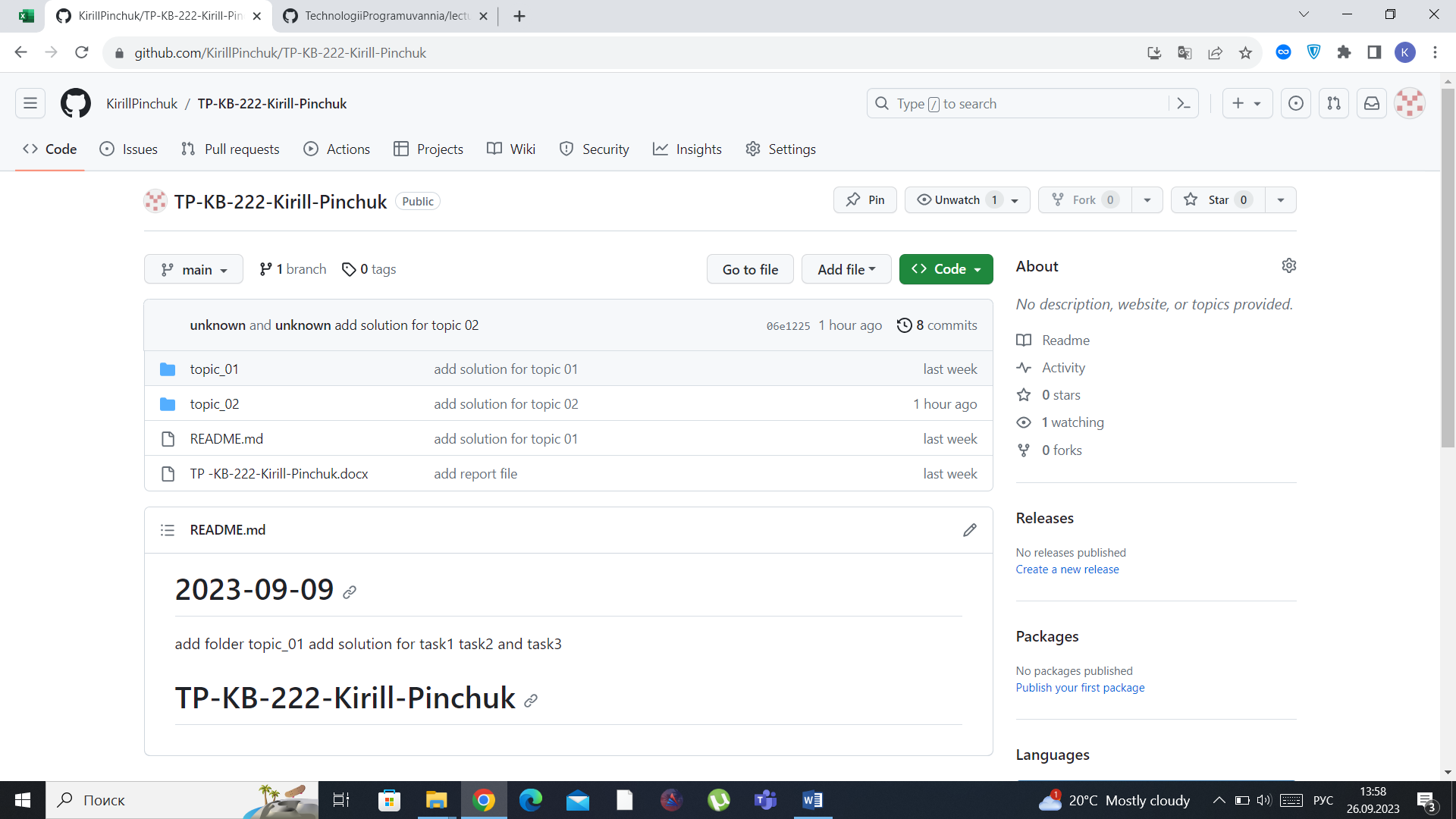
print(result)

else:

print("Invalid operation")

Посилання на github: **https://github.com/KirillPinchuk**

Знімок екрану з посилання на github:



**Звіт до Теми №3**

Функції та змінні

Під час виконання практичного завдання до Теми №3було надано варіанти рішення до наступних задач:

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.
2. Написати програму тестування функцій списків таких як: extend(), append(), insert(id, val), remove(val), clear(), sort(), reverse(), copy()
3. Написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items()
4. Маючи відсортований список, написати функцію пошуку позиції для вставки нового елементу в список.

**Хід виконання завдання:**

**1.** Програма запускається у циклі while True, що дозволяє безкінечно повторювати запити, поки користувач не введе правильну операцію. Визначаються функції sum, minus, multiplication, та division, які виконують відповідні математичні операції за допомогою перевірки "OP". Якщо операція введена неправильно (не "+", "-", "\*", або "/"), програма виводить повідомлення про неправильну операцію і повторює запит. Цикл while продовжується, поки користувач не введе правильну операцію. Після виведення результату програма завершує роботу.

**2.** Використання extend для додавання рядка до списку:

Починаю зі списку list, який має значення ["Emma", "Jon", "Bob"].

Використовуєте метод extend, щоб додати рядок "Liza" до списку.

Результат виводу у список ["Emma", "Jon", "Bob", "L", "i", "z", "a"].

Використання append для додавання елементу до списку:

Список list, який має значення ["Emma", "Jon", "Bob"].

Використовуєте метод append, щоб додати рядок "Liza" до списку.

Результат виводу у список ["Emma", "Jon", "Bob", "Liza"].

Використання insert для вставки елемента до списку за певним індексом:

Список list зі значеннями ["Emma", "Jon", "Bob"].

Застосовуєте метод insert, щоб вставити рядок "Liza" на позицію з індексом 1

Результат виводу у список ["Emma", "Liza", "Jon", "Bob"].

Використання remove для видалення конкретного елемента зі списку:

Список list зі значеннями ["Emma", "Jon", "Bob"].

Використовуєте метод remove, щоб видалити рядок "Jon" зі списку.

Результат виводу у список ["Emma", "Bob"].

Використання clear для очищення списку:

Список list зі значеннями ["Emma", "Jon", "Bob"].

Використовуєте метод clear, щоб очистити список.

Результат виводу у порожній список [].

Використання sort для сортування списку:

Список list зі значеннями ["Emma", "Jon", "Bob"].

Використовуєте метод sort, щоб відсортувати список в алфавітному порядку.

Результат виводу у список ["Bob", "Emma", "Jon"].

Використання reverse для обертання списку:

Список list зі значеннями ["Emma", "Jon", "Bob"].

Використовуєте метод reverse, щоб обернути порядок елементів у списку.

Результат виводу у список ["Bob", "Jon", "Emma"].

Список list зі значеннями ["Emma", "Jon", "Bob"].

Використовуєте метод copy, щоб створити копію списку, яка має ті ж самі значення.

Результат виводу копії списку ["Emma", "Jon", "Bob"].

**3.** Вивожу початковий словник dict з ключами "name", "age", "city" і відповідними значеннями.

Вивожу оригінальний словник, використовуючи print("Original =", dict).

Додавання нового ключа та значення до словника:

Використовую метод update, щоб додати новий ключ "country" і його значення "USA" до словника.

Виводите словник після оновлення, використовуючи print(dict).

Видалення ключа та відповідного значення зі словника:

Використовую оператор del, щоб видалити ключ "age" та відповідне значення зі словника.

Виводите словник після видалення, використовуючи print(dict).

Отримання списку ключів словника:

Використовую метод keys(), щоб отримати список ключів словника.

Виводите список ключів, використовуючи print(keys).

Отримання списку значень словника:

Використовую метод values(), щоб отримати список значень словника.

Виводите список значень, використовуючи print(values).

Отримання списку пар ключ-значення словника:

Використовую метод items(), щоб отримати список кортежів, кожний з яких містить пару ключ-значення словника.

Виводите список пар, використовуючи print(items).

Очищення словника:

Використовую метод clear(), щоб видалити всі ключі та значення зі словника.

Виводите словник після очищення, використовуючи print(dict).

Це послідовність дій для кожного з операцій, які ви виконуєте зі словником dict.

**4.** Спочатку, ми маємо функцію InsterToPosition, яка приймає відсортований список sorted\_list та новий елемент NewValue. У функції ви створюєте змінну position і встановлюєте її значення на 0. Запускаю цикл for, який проходиться по елементах у відсортованому списку sorted\_list.

У циклі перевіряю, чи NewValue менше за поточний елемент i в sorted\_list. Якщо це так, то ви збільшуєте значення position на 1. Після завершення циклу повертаю значення position, яке вказує на позицію, на яку слід вставити NewValue в список sorted\_list.

Викликаю функцію InsterToPosition, передаючи їй відсортований список list і NewValue, і зберігаєте результат у змінній insert\_position. Встановлюю NewValue в список list на позицію insert\_position за допомогою методу insert().

Вивожу позицію вставки (insert\_position) і оновлений список list за допомогою функції print().

**Текст програми:**

1. while True:

a = int(input("Enter a: "))

b = int(input("Enter b: "))

op = input("Operation: ")

def sum(a, b):

return a + b

def minus(a,b):

return a - b

def multiplication(a,b):

return a \* b

def division(a, b):

if b != 0:

return a / b

else:

print("Division by zero is not allowed")

if op == "+":

result = sum(a,b)

break

elif op == "-":

result = minus(a,b)

break

elif op == "\*":

result = multiplication(a,b)

break

elif op == "/":

result = division(a,b!=0)

break

print("result =", result)

**2.** list = ["Emma", "Jon", "Bob"]

list.extend("Liza")

print(list)

list = ["Emma", "Jon", "Bob"]

list.append("Liza")

print(list)

list = ["Emma", "Jon", "Bob"]

list.insert(1, "Liza")

print(list)

list = ["Emma", "Jon", "Bob"]

list.remove("Jon")

print(list)

list = ["Emma", "Jon", "Bob"]

list.clear()

print(list)

list = ["Emma", "Jon", "Bob"]

list.sort()

print(list)

list = ["Emma", "Jon", "Bob"]

list.reverse()

print(list)

list = ["Emma", "Jon", "Bob"]

list.copy()

print(list)

**3**. dict = {"name": "John", "age": 30, "city": "New York"}

print("Original =", str)

dict.update({"country": "USA"})

print(dict)

del dict ["age"]

print(dict)

keys = dict.keys()

print(keys)

values = dict.values()

print(values)

items = dict.items()

print(items)

dict.clear()

print(dict)

**4.** def InsterToPosition(sorted\_list, NewValue):

position = 0

for i in sorted\_list:

if NewValue < i:

position +=1

return position

list = ["aa", "ee", "zz"]

NewValue = input("New value: ")

insert\_position = InsterToPosition(list, NewValue)

list.insert(insert\_position, NewValue)

print(f"position: {insert\_position}, list: {list}")

Посилання на github: <https://github.com/KirillPinchuk>

Знімок екрану з посилання на github: 