**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: Повернення рядка в зворотному порядку, Тестування базових функцій для рядків, Написання функції пошуку Дискримінанту.

task1

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

Текст програми:

str1="abcdefg123"

str2=""

for i in **range**(10):

    str2+=str1[9-i]

**print** (str2)

task2

Необхідно протестувати базові функції для рядків.

Текст програми:

x = "QwErTy"

a = x.**strip**()

b = x.**capitalize**()

c = x.**title**()

d = x.**upper**()

e = x.**lower**()

**print** (a,"\n",b,"\n",c,"\n",d,"\n",e,"\n")

task3

Необхідно написати функцію пошуку дискримінанту.

Текст програми:

import math

def **discr**(a,b,c):

    return b\*\*2 - (4\*a\*c)

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

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

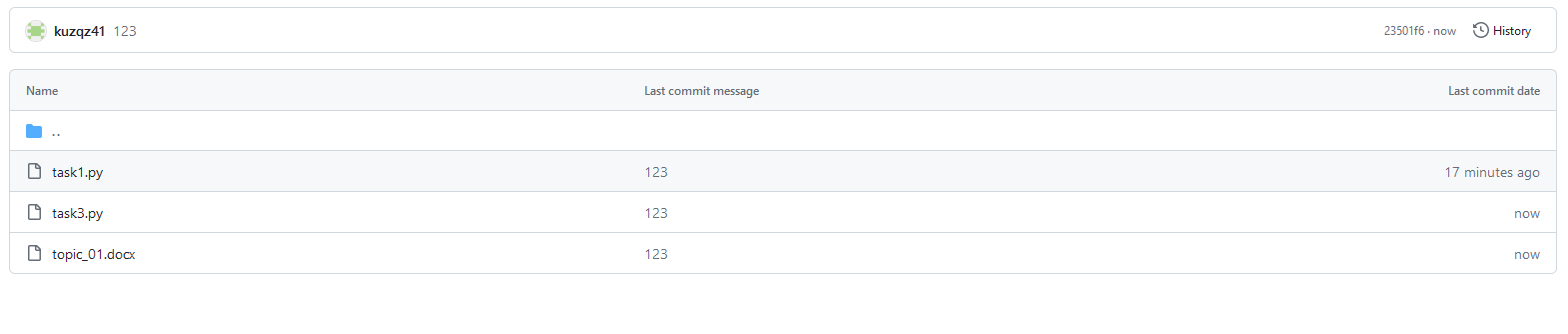
c = int (**input**("c: "))

d = discr(a,b,c)

**print**("d= ", d)

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_01>

Знімок екрану з посилання на github:



Звіт до Теми №2

Функції перевірки

Під час виконання практичного завдання до Теми №2 було надано варіанти рішення до наступних задач: пошук коренів квадратного рівняння, створення калькуляторів за допомогою if, elif, else та match.

task1

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

Текст програми:

import math

def **discr**(a,b,c):

    return b\*\*2 - (4\*a\*c)

def **roots**(a,b,c):

    d = discr(a,b,c)

    if d > 0:

        x1 = (-b + math.sqrt(d))/2\*a

        x2 = (-b - math.sqrt(d))/2\*a

    elif d == 0:

        x1 = (-b + math.sqrt(d))/2\*a

        x2 = "Кореня не існує"

    else:

        x1 = "Кореня не існує"

        x2 = "Кореня не існує"

    return x1, x2

**print** ("Квадратне рівняння ax^2 + bx + c = 0")

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

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

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

d = discr(a,b,c);

**print** ("d= ",d)

x1, x2 = roots(a,b,c)

**print** ("x1= ",x1,"      x2= ",x2)

task2

Написати програму калькулятор використовуючи **if else** конструкцію

Текст програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

a = **int**(**input** ("Введіть перше число: "))

b = **int**(**input** ("Введіть друге число: "))

x = **str**(**input**("Введіть знак (+ або - або \* або /): "))

if x=="+":

**print** ("a + b = ", **sum**(a,b))

elif x=="-":

**print** ("a - b = ", **minus**(a,b))

elif x=="\*":

**print** ("a \* b = ", **mult**(a,b))

elif x=="/":

    if b==0:

**print** ("Ділити на нуль не можна")

    else:

**print** ("a / b = ", **div**(a,b))

else:

**print** ("Ви ввели неправильний знак")

task3

Написати програму калькулятор використовуючи **match** конструкцію

Текст програми:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

a = **int**(**input** ("Введіть перше число: "))

b = **int**(**input** ("Введіть друге число: "))

x = **str**(**input**("Введіть знак (+ або - або \* або /): "))

match x:

    case "+":

**print** ("a + b = ",**sum**(a,b))

    case "-":

**print** ("a - b = ",**minus**(a,b))

    case "\*":

**print** ("a \* b = ",**mult**(a,b))

    case "/":

        match b:

            case 0:

**print** ("Ділити на нуль не можна")

            case \_:

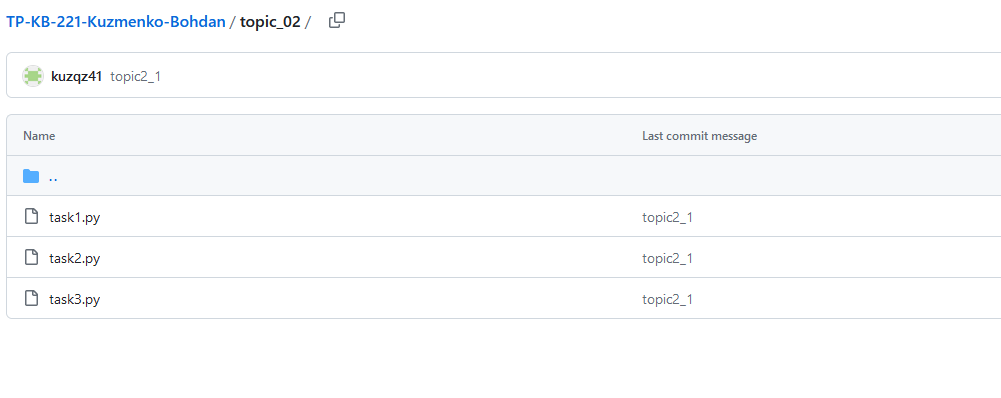
**print** ("a / b = ",**div**(a,b))

    case \_:

**print** ("Ви ввели неправильний знак")

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_02>

Знімок екрану з посилання на github:



Звіт до Теми №3

Цикли

Під час виконання практичного завдання до Теми №3 було надано варіанти рішення до наступних задач: Написання калькулятора з постійними запитами на введення даних, Тестування функцій для списків, Тестування функцій для словників, Написання функції пошуку позиції елементу у відсортованому списку.

task1

Необхідно 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:

def **sum**(a,b):

    return a + b

def **minus**(a,b):

    return a - b

def **mult**(a,b):

    return a \* b

def **div**(a,b):

    return a / b

while 1:

    a = **int**(**input** ("Введіть перше число: "))

    b = **int**(**input** ("Введіть друге число: "))

    x = **str**(**input**("Введіть знак (+ або - або \* або /) (exit щоб вийти): "))

    match x:

        case "+":

**print** ("a + b = ",**sum**(a,b))

        case "-":

**print** ("a - b = ",**minus**(a,b))

        case "\*":

**print** ("a \* b = ",**mult**(a,b))

        case "/":

            match b:

                case 0:

**print** ("Ділити на нуль не можна")

                case \_:

**print** ("a / b = ",**div**(a,b))

        case "exit":

            break

        case \_:

**print** ("Ви ввели неправильний знак")

task2

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

Текст програми:

a=[1,2,3]

b=[6,2,5]

c=[]

d=[]

a.**extend**(b)

**print** ("extend - ",a)

c.**append**(1)

c.**append**(9)

**print** ("append - ",c)

a.**insert**(2, "qwe")

**print** ("insert - ",a)

a.**remove** (3)

**print** ("remove - ",a)

a.**clear**()

**print** ("clear - ",a)

b.**sort**()

**print**("sort - ",b)

b.**reverse**()

**print**("reverse - ",b)

d = b.**copy**()

**print**("copy - ",d)

task3

Необхідно написати програму тестування функцій словників таких як: update(), del(), clear(), keys(), values(), items()

Текст програми:

a={'qwe':123}

b={'asd':456}

c={'qwe':123,'asd':456}

a.**update**(b)

**print** ("update - ",a)

del(b)

a.**clear**()

**print**("clear - ",a)

x=c.**keys**()

**print** ("keys - ",x)

x=c.**values**()

**print** ("values - ",x)

x=c.**items**()

**print**("items - ",x)

task4

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

Текст програми:

def **PosSearch**(a,b):

    for i in **range**(**len**(a)):

        if a[i] == b:

            return i

a = ["aa", "bb", "cc", "dd", "ee", "ff", "gg", "hh", "ii", "jj", "kk", "ll", "mm", "nn", "oo", "pp", "qq", "rr", "ss", "tt", "uu", "vv", "ww", "xx", "yy", "zz"]

b = **str**(**input**("b: "))

a.**append**(b)

a.**sort**()

**print** (a)

i = **PosSearch** (a,b)

**print** ("Елемент ", b," знаходиться на ",i," позиції")

Посилання на github: <https://github.com/kuzqz41/TP-KB-221-Kuzmenko-Bohdan/tree/main/topic_03>