**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:

Текст програми:

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

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_01/file_01.py>

Знімок екрану з посилання на github:



**Тест**

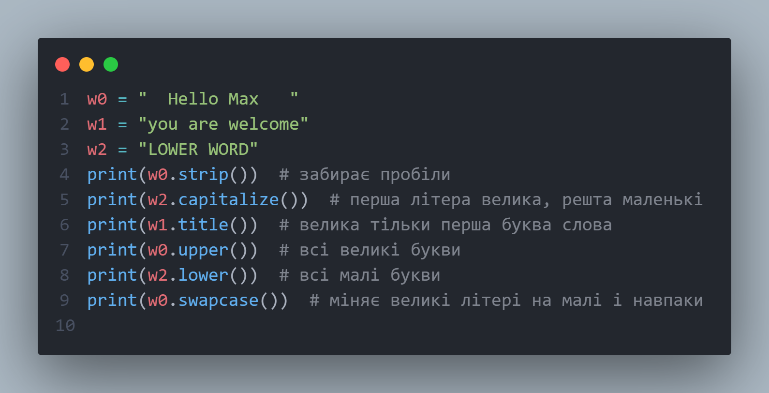
Виконати деякі тести для strip, capitalize, title, upper, lower та деяких інших Хід виконання завдання:

Текст програми:

|  |
| --- |
| w0 = " Hello Max "  w1 = "you are welcome"  w2 = "LOWER WORD"  print(w0.strip())  print(w2.capitalize())  print(w1.title())  print(w0.upper())  print(w2.lower())  print(w0.swapcase()) |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_01/file_02.py>

Знімок екрану з посилання на github:



**Функція пошуку Дискримінант**

Написати формулу дискримінанту і знайти його.

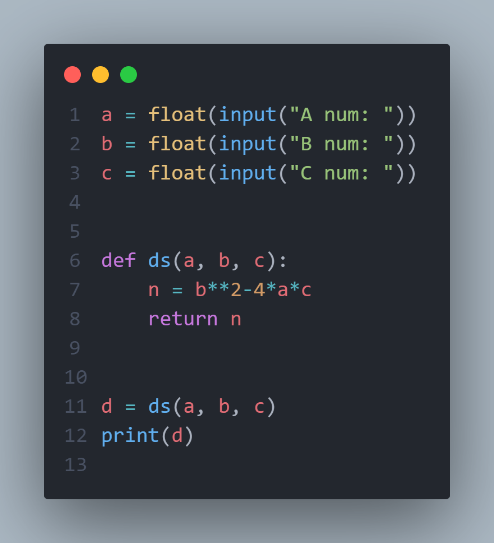
Хід виконання завдання:

Текст програми:

|  |
| --- |
| a = float(input("A num: "))  b = float(input("B num: "))  c = float(input("C num: "))  def ds(a, b, c):  n = b\*\*2-4\*a\*c  return n  d = ds(a, b, c)  print(d) |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_01/file_03.py>

Знімок екрану з посилання на github:



**Пошуку коренів квадратного рівняння**

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

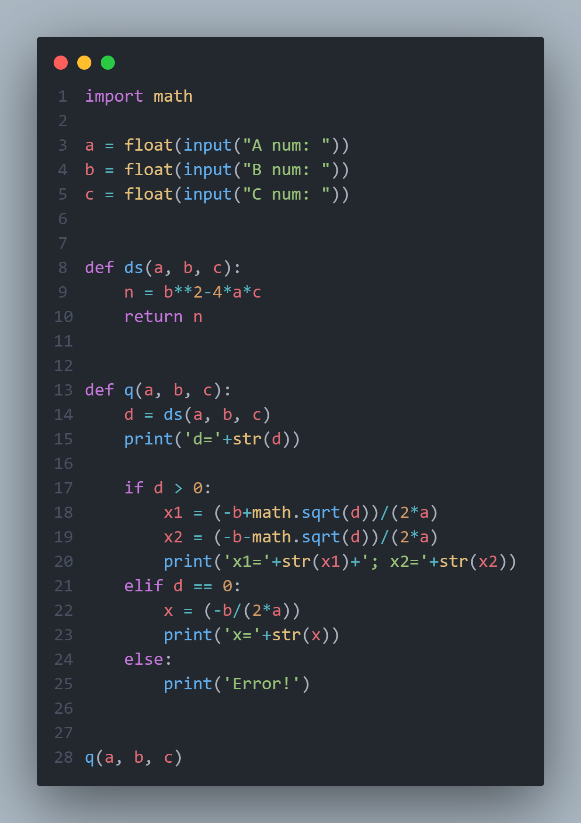
Хід виконання завдання:

Текст програми:

|  |
| --- |
| import math  a = float(input("A num: "))  b = float(input("B num: "))  c = float(input("C num: "))  def ds(a, b, c):      n = b\*\*2-4\*a\*c      return n  def q(a, b, c):      d = ds(a, b, c)      print('d='+str(d))      if d > 0:          x1 = (-b+math.sqrt(d))/(2\*a)          x2 = (-b-math.sqrt(d))/(2\*a)          print('x1='+str(x1)+'; x2='+str(x2))      elif d == 0:          x = (-b/(2\*a))          print('x='+str(x))      else:          print('Error!')  q(a, b, c) |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_02/file_01.py>

Знімок екрану з посилання на github:



**Калькулятор через if else**

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

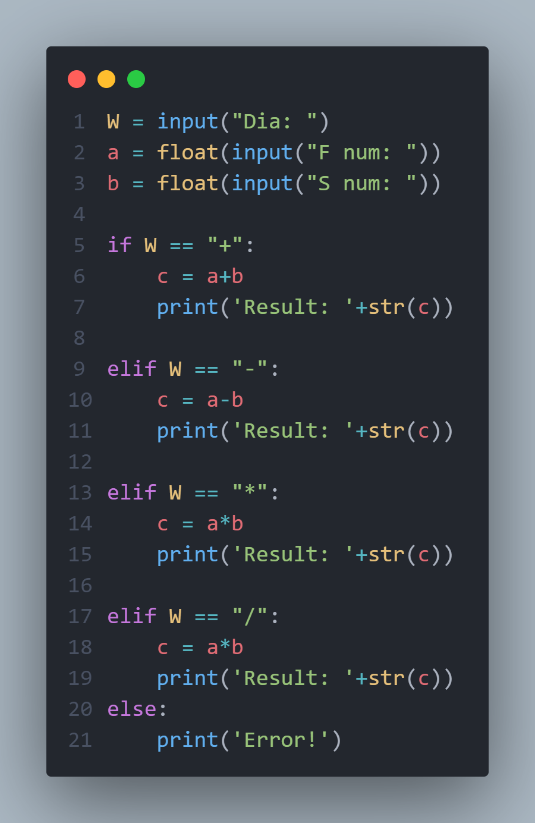
Хід виконання завдання:

Текст програми:

|  |
| --- |
| W = input("Dia: ")  a = float(input("F num: "))  b = float(input("S num: "))  if W == "+":      c = a+b      print('Result: '+str(c))  elif W == "-":      c = a-b      print('Result: '+str(c))  elif W == "\*":      c = a\*b      print('Result: '+str(c))  elif W == "/":      c = a\*b      print('Result: '+str(c))  else:      print('Error!') |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_02/file_02.py>

Знімок екрану з посилання на github:



**Калькулятор на match**

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

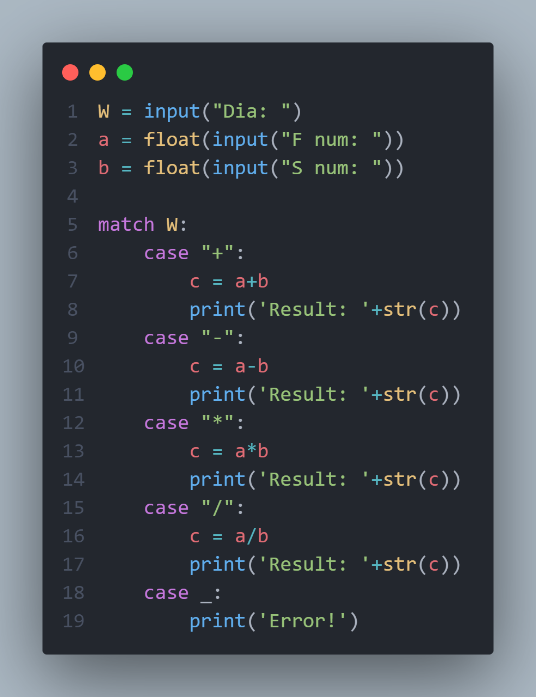
Хід виконання завдання:

Текст програми:

|  |
| --- |
| W = input("Dia: ")  a = float(input("F num: "))  b = float(input("S num: "))  match W:      case "+":          c = a+b          print('Result: '+str(c))      case "-":          c = a-b          print('Result: '+str(c))      case "\*":          c = a\*b          print('Result: '+str(c))      case "/":          c = a/b          print('Result: '+str(c))      case \_:          print('Error!') |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_02/file_03.py>

Знімок екрану з посилання на github:



**Програму калькулятор з постійними запитами**

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.

Хід виконання завдання:

Текст програми:

|  |
| --- |
| while True:      print('--------------------------------')      W = input("Dia: ")      if W == "ex":          print('exit')          break      a = float(input("F num: "))      b = float(input("S num: "))      match W:          case "+":              c = a+b              print('Result: '+str(c))          case "-":              c = a-b              print('Result: '+str(c))          case "\*":              c = a\*b              print('Result: '+str(c))          case "/":              c = a/b              print('Result: '+str(c))          case \_:              print('Error!') |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_03/file_01.py>

Знімок екрану з посилання на github:



**Тестування функцій списків**

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

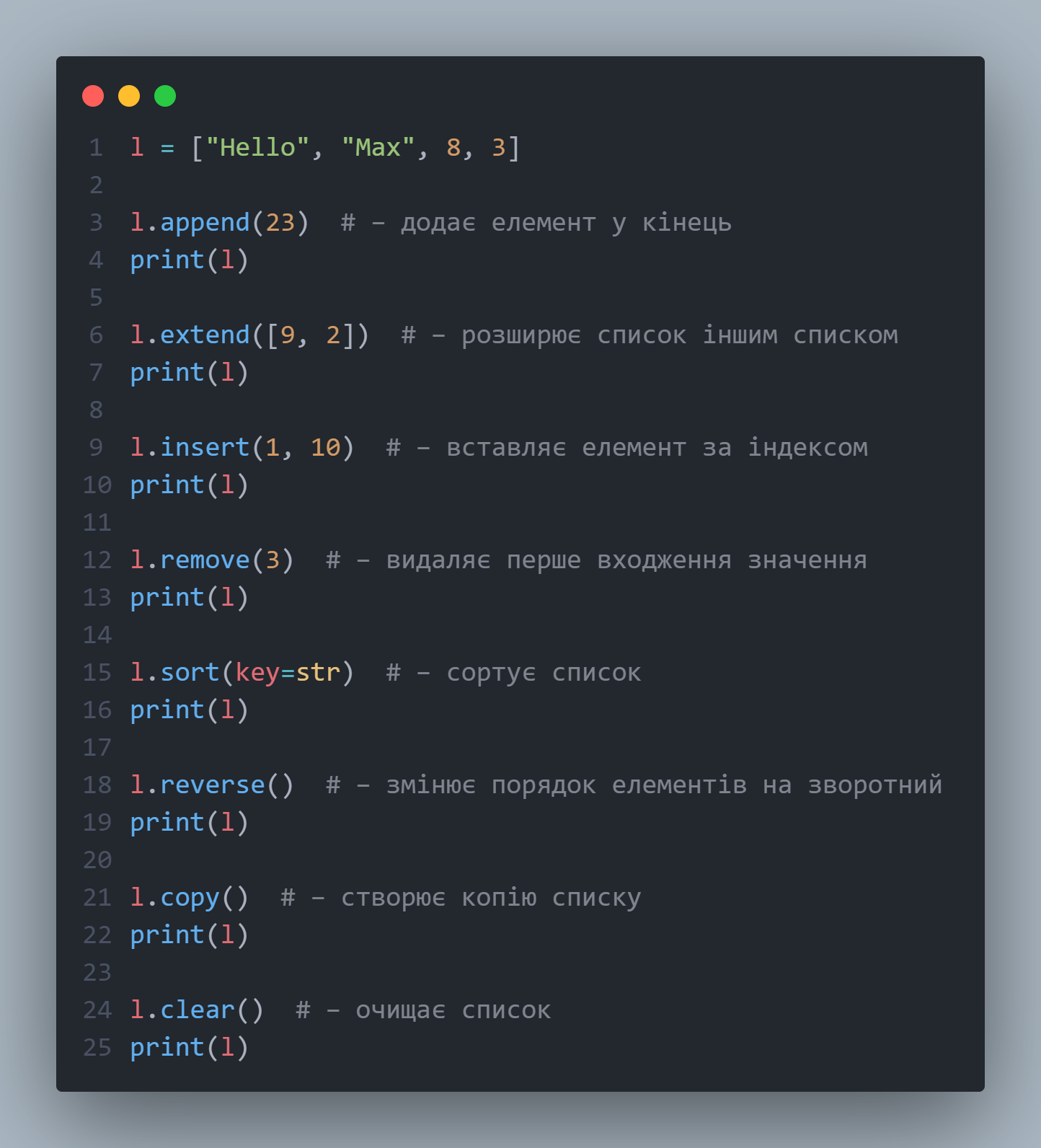
Хід виконання завдання:

Текст програми:

|  |
| --- |
| l = ["Hello", "Max", 8, 3]  l.append(23)  # – додає елемент у кінець  print(l)  l.extend([9, 2])  # – розширює список іншим списком  print(l)  l.insert(1, 10)  # – вставляє елемент за індексом  print(l)  l.remove(3)  # – видаляє перше входження значення  print(l)  l.sort(key=str)  # – сортує список  print(l)  l.reverse()  # – змінює порядок елементів на зворотний  print(l)  l.copy()  # – створює копію списку  print(l)  l.clear()  # – очищає список  print(l) |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_03/file_02.py>

Знімок екрану з посилання на github:



**Тестування функцій словників**

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

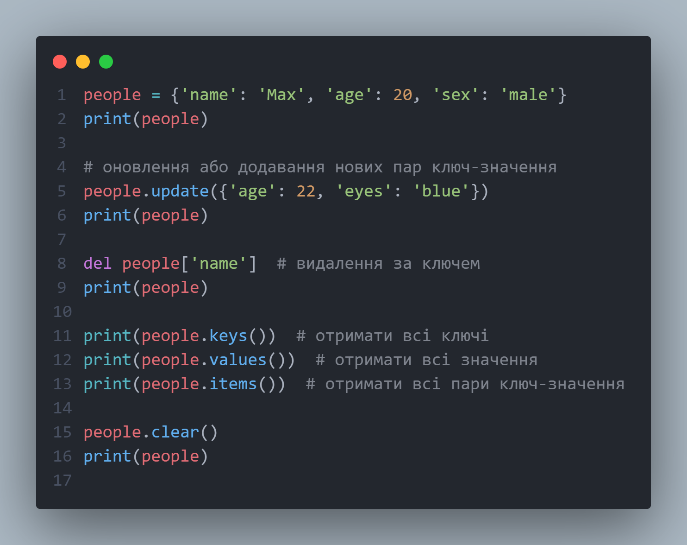
Хід виконання завдання:

Текст програми:

|  |
| --- |
| people = {'name': 'Max', 'age': 20, 'sex': 'male'}  print(people)  # оновлення або додавання нових пар ключ-значення  people.update({'age': 22, 'eyes': 'blue'})  print(people)  del people['name']  # видалення за ключем  print(people)  print(people.keys())  # отримати всі ключі  print(people.values())  # отримати всі значення  print(people.items())  # отримати всі пари ключ-значення  people.clear()  print(people) |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_03/file_03.py>

Знімок екрану з посилання на github:



**Пошуку позиції для вставки**

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

Хід виконання завдання:

Текст програми:

|  |
| --- |
| type\_list = [3, 2, 6, 8, 13, 76, 5]  type\_list.sort()  print(type\_list)  num = int(input('What is your number: '))  def vstavka(type\_list, num):      for i, e in enumerate(type\_list):          if num <= e:              return i      return len(type\_list)  pos = vstavka(type\_list, num)  print('Your position: ', pos) |

Посилання на github: <https://github.com/mr1inux/TP-KB-242-KORETSKYI-M/blob/main/topic_03/file_04.py>

Знімок екрану з посилання на github:

