Івано-Франківський національний технічний університет нафти і газу

Кафедра комп’ютерних систем і мереж

**ЛАБОРАТОРНА РОБОТА №5**

з дисципліни Практикум Python

Тема: Розробка програм з використанням процедур і

функцій

Виконав:

Ст. гр. КІ-23-2 (Боднар Р.В.)

Перевірив: (Кропивницький Д. Р.)

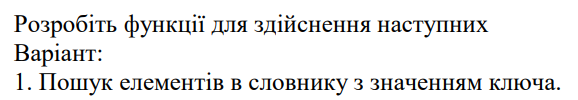
м. Івано-Франківськ

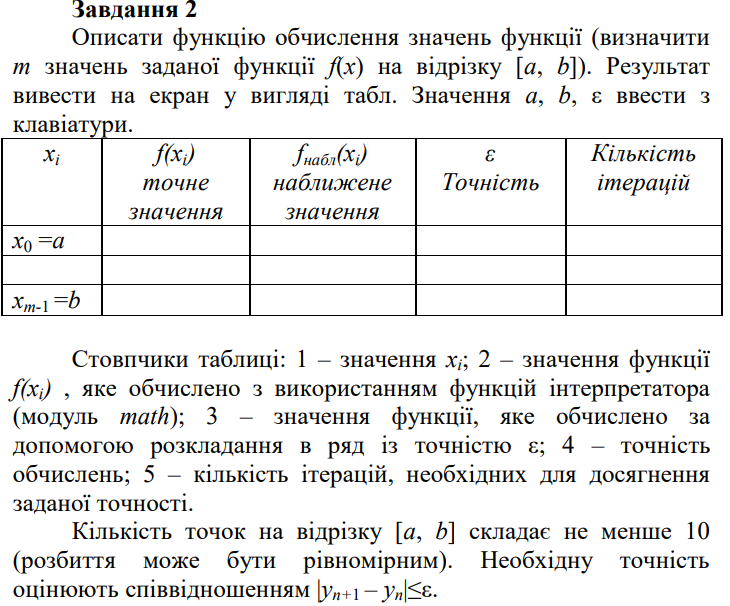
2025

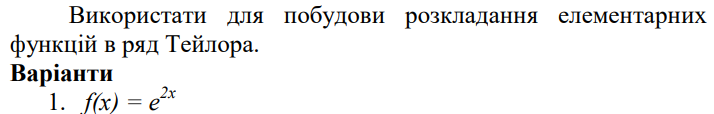
**Мета роботи:** Познайомитися з принципами побудови функцій користувача на мові Python, з використанням локальних і глобальних змінних.

**Варіант 1**

**Завдання:**









**Хід роботи**

**Код програми до завдання 1:**

def find\_key(d, key):

    return d.get(key, "Key not found")

def find\_value(lst, value):

    return lst.index(value) if value in lst else "Value not found"

def find\_sequence\_in\_list(lst, subseq):

    for i in range(len(lst) - len(subseq) + 1):

        if lst[i:i+len(subseq)] == subseq:

            return i

    return "Sequence not found"

def first\_five\_minimal(lst):

    return sorted(lst)[:5]

def first\_five\_maximal(lst):

    return sorted(lst, reverse=True)[:5]

def average\_value(lst):

    return sum(lst) / len(lst) if lst else 0

def find\_zero\_values(lst):

    return [i for i, v in enumerate(lst) if v == 0]

def list\_without\_duplicates(lst):

    result = []

    for x in lst:

        if x not in result:

            result.append(x)

    return result

def find\_sequence\_in\_string(s, subseq):

    idx = s.find(subseq)

    return idx if idx != -1 else "Sequence not found"

def split\_sentence(sentence):

    return sentence.split()

dictionary = {'a': 1, 'b': 2, 'c': 3}

lst = [1, 0, 3, 5, 0, 2, 7, 5, 3, 0]

sentence = "lorem ipsum dolor sit amet,   consectetur  adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."

while True:

    print("\n=== MENU ===")

    print("1. Search in dictionary by key")

    print("2. Search element by value in list")

    print("3. Search sequence of elements in list")

    print("4. First 5 minimal elements in list")

    print("5. First 5 maximal elements in list")

    print("6. Average value in list")

    print("7. All zero elements in list (indices)")

    print("8. List without duplicates")

    print("9. Search sequence in string")

    print("10. Split sentence into words")

    print("0. Exit")

    choice = input("Your choice: ")

    if choice == "0":

        break

    elif choice == "1":

        key = input("Enter key: ")

        print("Result:", find\_key(dictionary, key))

    elif choice == "2":

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

        print("Result:", find\_value(lst, value))

    elif choice == "3":

        subseq = list(map(int, input("Enter sequence (space separated): ").split()))

        print("Result:", find\_sequence\_in\_list(lst, subseq))

    elif choice == "4":

        print("Result:", first\_five\_minimal(lst))

    elif choice == "5":

        print("Result:", first\_five\_maximal(lst))

    elif choice == "6":

        print("Result:", average\_value(lst))

    elif choice == "7":

        print("Result (indices of zeros):", find\_zero\_values(lst))

    elif choice == "8":

        print("Result:", list\_without\_duplicates(lst))

    elif choice == "9":

        subseq = input("Enter substring: ")

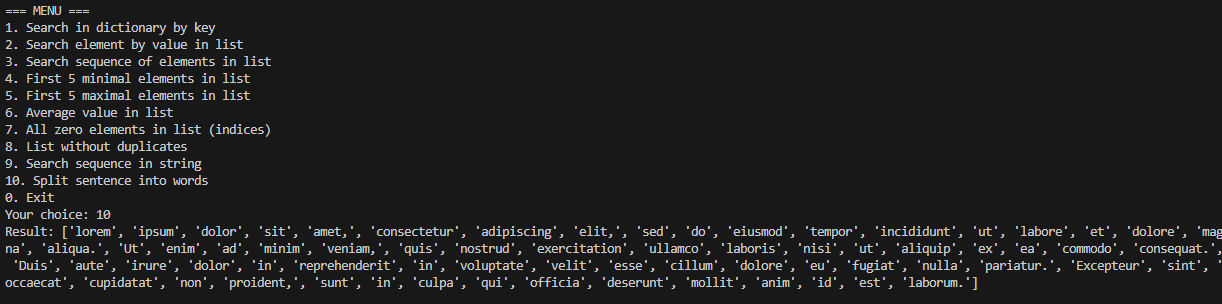
        print("Result:", find\_sequence\_in\_string(sentence, subseq))

    elif choice == "10":

        print("Result:", split\_sentence(sentence))

    else:

        print("Invalid choice, try again.")

****

**Рисунок 1 - Результат виконання завдання 1**

**Код програми до завдання 2:**

import math

m = [1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5]

a = 0

b = 7

eps = 1e-8

def taylor\_series(x, eps):

    n = 0

    term = 1

    s = term

    while True:

        n += 1

        term = (2 \* x) \*\* n / math.factorial(n)

        s\_new = s + term

        if abs(s\_new - s) <= eps:

            return s\_new, n

        s = s\_new

h = (b - a) / (len(m) - 1)

print(f"{'x':>10} | {'f(x) точне':>15} | {'f(x) наближене':>20} | {'похибка':>15} | {'ітерацій':>10}")

print("-" \* 80)

for i in range(len(m)):

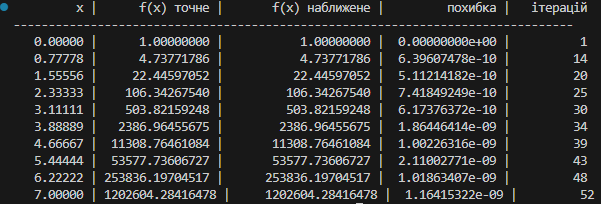
    x = a + i \* h

    exact = math.exp(2 \* x)

    approx, iterations = taylor\_series(x, eps)

    error = abs(exact - approx)

    print(f"{x:10.5f} | {exact:15.8f} | {approx:20.8f} | {error:15.8e} | {iterations:10d}")

****

**Рисунок 2 - Результат виконання завдання 2**

**Висновок:** У роботі реалізовано програми з використанням власних функцій. Перша – функції для роботи зі словниками, списками та реченнями, друга – для обчислення заданої функції на відрізку [a, b] за допомогою ряду Тейлора з контролем точності.