МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"

Кафедра інформаційних систем та мереж

Лабораторна робота №1

з курсу

СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ

Виконав студент

групи ІТ-21сп

Дубов М.А.

Прийняв

Щербак С.С.

Львів - 2023

**Лабораторна робота №1**

**Мета:** створення консольної програми-калькулятора за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.

**Хід роботи**

**Завдання 1. Введення користувача**

Створити Python-програму, яка приймає введення користувача для двох чисел і оператора (наприклад, +, -, \*, /).

firstInput = float(input("Enter the first number: "))

secondInput = float(input("Enter the second number: "))

operator = input("Enter the operator symbol: ")

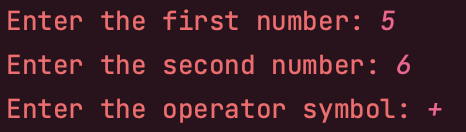


Рис 1. Виконання завдання 1

**Завдання 2. Перевірка оператора.**

Перевірити чи введений оператор є дійсним (тобто одним із +, -, \*, /). Якщо ні, відобразити повідомлення про помилку і попросити користувача ввести дійсний оператор.

def check\_correct\_operator(o):

operators = ['+', '-', '\*', '/', '^', 'sqrt', '%']

if o not in operators:

print("Invalid operator entered, please try again")

arithmetic\_calculator()

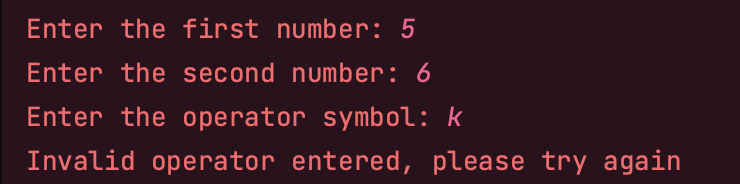


Рис 2. Виконання завдання 2

**Завдання 3. Обчислення.**

Виконати обчислення на основі введення користувача (наприклад, додавання, віднімання, множення, ділення) і відобразити результат.

def arithmetic\_calculator():

firstInput = float(input("Enter the first number: "))

secondInput = float(input("Enter the second number: "))

operator = input("Enter the operator symbol: ")

result = 0

error\_handler(firstInput, secondInput, operator)

check\_correct\_operator(operator)

if operator == '+':

result = firstInput + secondInput

elif operator == '-':

result = firstInput - secondInput

elif operator == '\*':

result = firstInput \* secondInput

elif operator == '/':

result = firstInput / secondInput

elif operator == '^':

result = pow(firstInput, secondInput)

elif operator == 'sqrt':

userChoice = int(input("this operator requires only one number. Press 1 or 2 for needed number"))

if userChoice == 1:

result = sqrt(firstInput)

elif userChoice == 2:

result = sqrt(secondInput)

else:

print("Wrong answer. You can only choose number one or two")

return

elif operator == '%':

result = firstInput % secondInput

result = int(round(result,decimalNumber))

print(f"{firstInput} {operator} {secondInput} = {result}")

archive\_expression(firstInput, operator, secondInput, result)

recursive\_calculation()

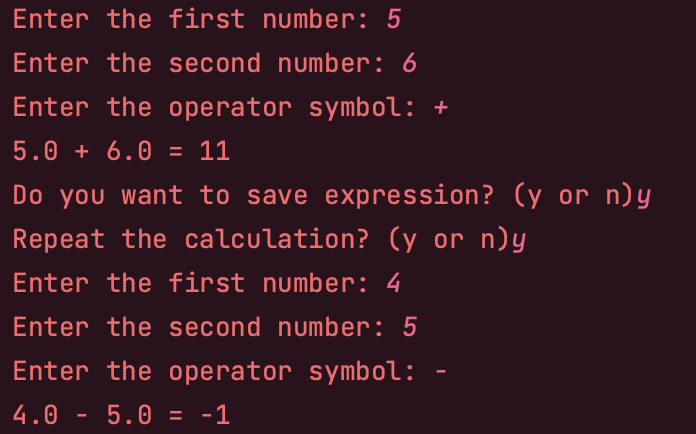


Рис 3. Виконання завдання 3

**Завдання 4. Повторення обчислень.**

Запитати користувача, чи він хоче виконати ще одне обчислення. Якщо так, дозволити йому ввести нові числа і оператор. Якщо ні, вийти з програми.

def recursive\_calculation():

answer = input("Repeat the calculation? (y or n)")

if answer == 'y':

arithmetic\_calculator()

recursive\_calculation()

elif answer == 'n':

main\_menu()

else:

print("Error: No such option. Please select y or n")

recursive\_calculation()

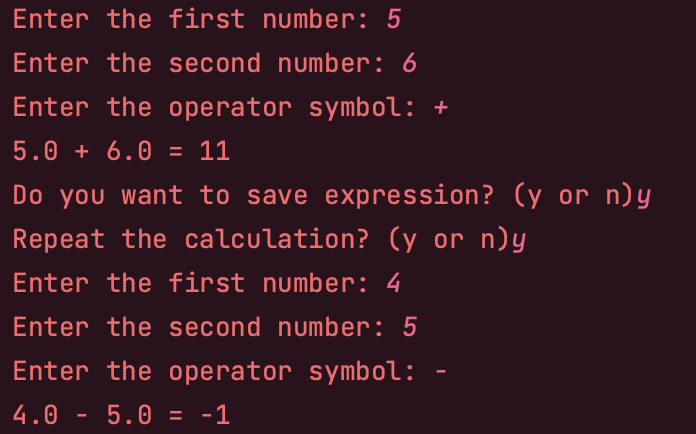


Рис 4. Виконання завдання 4

**Завдання 5. Обробка помилок.**

Реалізувати обробку помилок для обробки ділення на нуль або інших потенційних помилок. Відобразити відповідне повідомлення про помилку, якщо виникає помилка.

def error\_handler(numOne, numTwo, op):

if (numOne == 0) | (numTwo == 0) & (op == '/'):

print("Error: can't divide by zero")

recursive\_calculation()

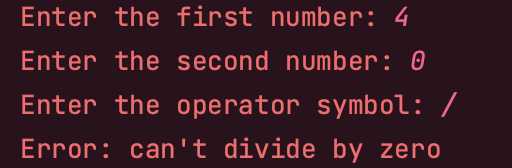


Рис 5. Виконання завдання 5

**Завдання 6. Десяткові числа.**

Змінити калькулятор так, щоб він обробляв десяткові числа (плаваючу кому) для більш точних обчислень.

Змінити калькулятор так, щоб він обробляв десяткові числа (плаваючу кому) для більш точних обчислень.

firstInput = float(input("Enter the first number: "))

secondInput = float(input("Enter the second number: "))

operator = input("Enter the operator symbol: ") firstInput = float(input("Enter the first number: "))

secondInput = float(input("Enter the second number: "))

operator = input("Enter the operator symbol: ")

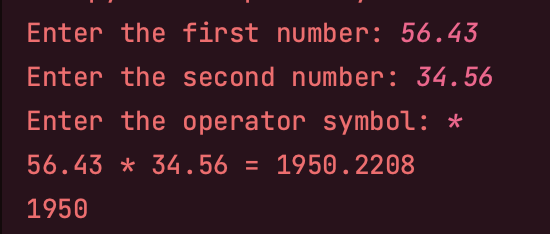
****

Рис 6. Виконання завдання 6

**Завдання 7. Додаткові операції.**

Додати підтримку додаткових операцій, таких як піднесення до степеня (^), квадратний корінь (√) і залишок від ділення (%).

elif operator == '^':

result = pow(firstInput, secondInput)

elif operator == 'sqrt':

userChoice = int(input("this operator requires only one number. Press 1 or 2 for needed number"))

if userChoice == 1:

result = sqrt(firstInput)

elif userChoice == 2:

result = sqrt(secondInput)

else:

print("Wrong answer. You can only choose number one or two")

recursive\_calculation()

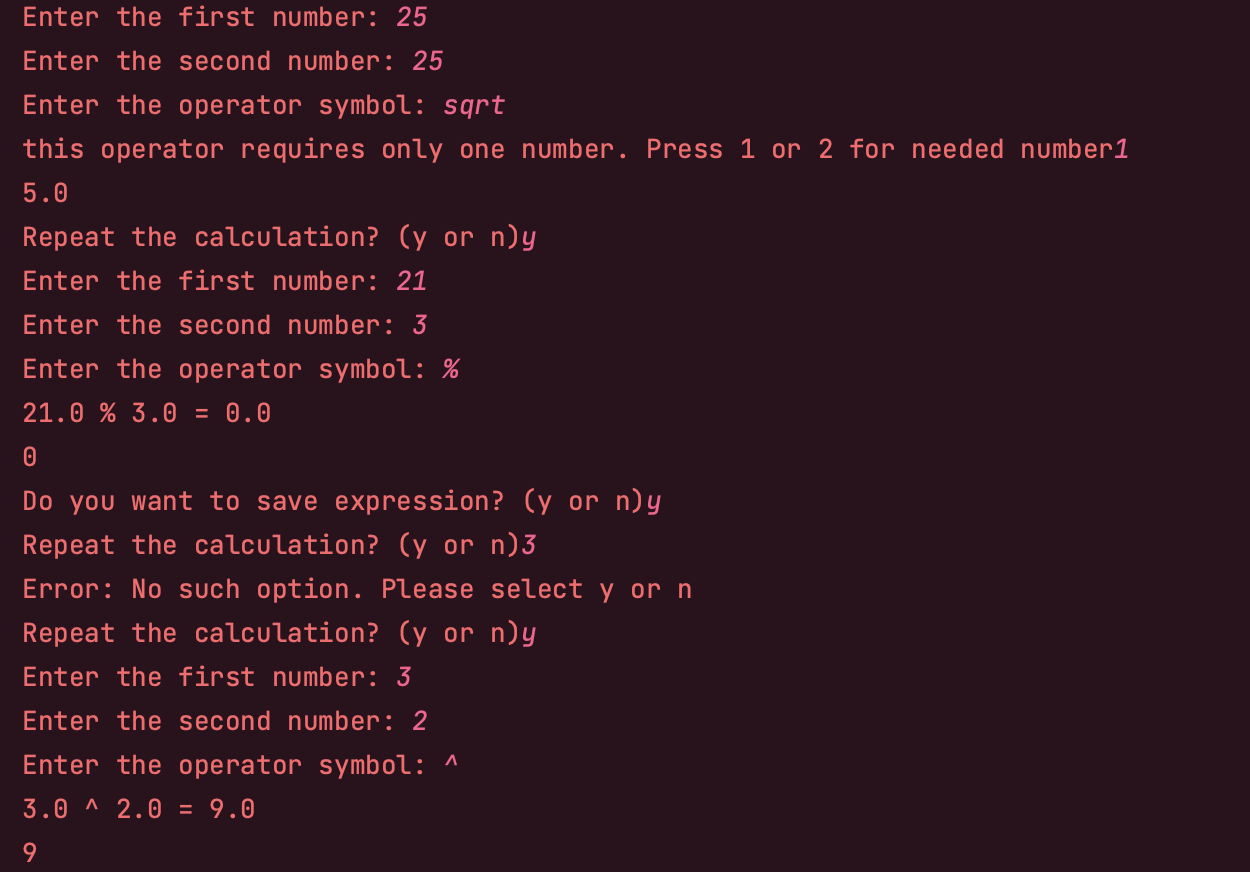


Рис 7. Виконання завдання 7

**Завдання 8. Функція пам'яті.**

Реалізувати функцію пам'яті, яка дозволяє користувачам зберігати і відновлювати результати. Додати можливості для зберігання та отримання значень з пам'яті.

**Завдання 9. Історія обчислень.**

Створити журнал, який зберігає історію попередніх обчислень, включаючи вираз і результат. Дозволити користувачам переглядати історію своїх обчислень.

def archive\_expression(firstNumber, operator, secondNumber, result):

expressionTemplate = f"{firstNumber} {operator} {secondNumber} = {result}"

userAnswer = input("Do you want to save expression? (y or n)")

if userAnswer == 'y':

archiveExpression.append(expressionTemplate)

recursive\_calculation()

if userAnswer == 'n':

recursive\_calculation()

def show\_archived\_expressions():

if len(archiveExpression) == 0:

print("Archive is empty!")

arithmetic\_calculator()

for savedExpressions in archiveExpression:

print(savedExpressions)

main\_menu()

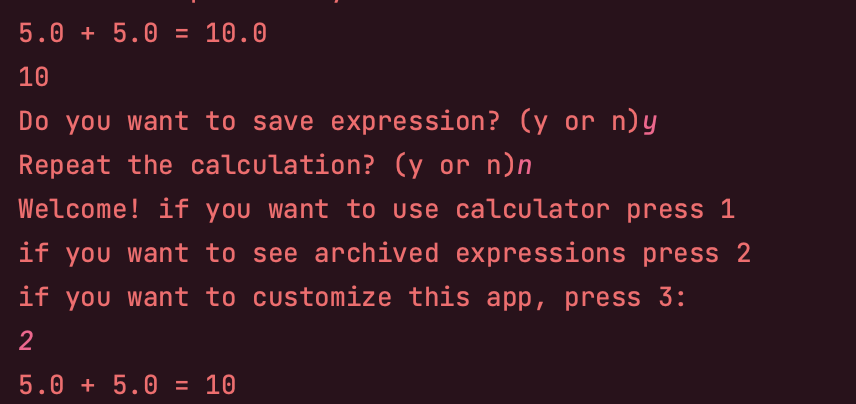


Рис 8. Виконання завдання 8 і 9

**Завдання 10. Налаштування користувача.**

Надати користувачам можливість налаштувати поведінку калькулятора, таку як зміну кількості десяткових розрядів, які відображаються, або налаштування функцій пам'яті.

Було додано можливість користувачу обирати кількість десяткових розрядів та очищувати історію обчислень.

def show\_settings():

global decimalNumber

print("if you want to delete archive of expressions press 1")

print("змінити заокруглення натисніть 2: ")

userInput = input()

if userInput == '1':

archiveExpression.clear()

main\_menu()

elif userInput == '2':

decimalNumber = int(input("введіть потрібне число: "))

if ((decimalNumber > 10) or (decimalNumber < 0)):

print("Incorrect number! Enter valid")

else:

print("заокруглення пройшло успішно")

arithmetic\_calculator()

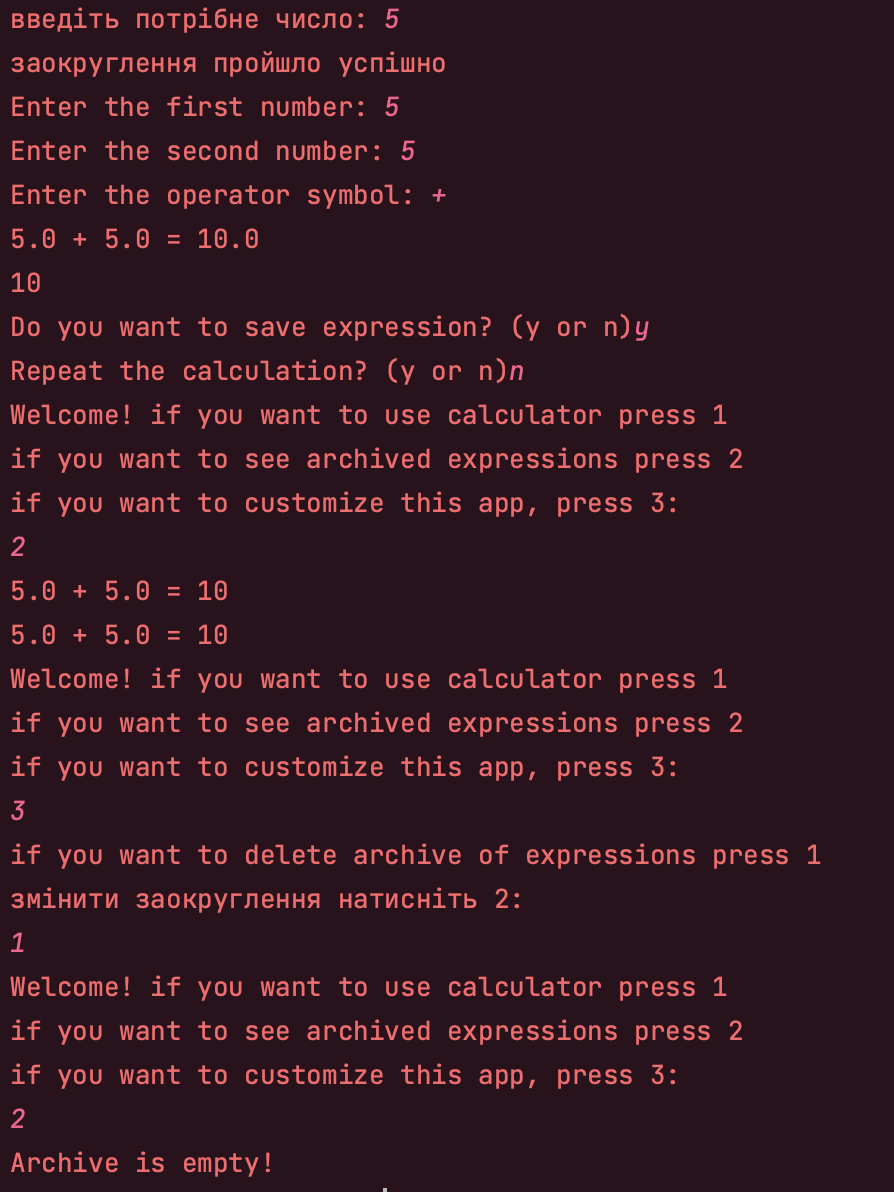


Рис 9. Виконання завдання 10

**Висновок:** під час опрацювання лабораторної роботи я навчився створювати консольну програму-калькулятор за допомогою основних синтаксичних конструкцій Python, з іншим завданням на заміну тестуванню та валідації.