МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського

«Харківський авіаційний інститут»

Кафедра систем управління літальними апаратами

Лабораторна робота № 1

з дисципліни «Об'єктно-орієнтоване програмування СУ»

Тема: «Розробка програм для математичних обчислень в Python»

ХАІ.301 . 174. 312ст.1 ЛР

Виконав студент гр.\_\_\_\_\_\_312ст\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Твердохліб Максим\_

(підпис, дата) (П.І.Б.)

Перевірив

\_\_\_\_\_\_\_\_\_\_\_\_\_\_к.т.н.,доц. О. В.Гавриленко

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ас.  В. О.Білозерський (підпис, дата) (П.І.Б.)

2024

# МЕТА РОБОТИ

Ознайомитися з основами розробки інтерпретованих програм і реалізувати скрипт для введення / виведення даних і виконання найпростіших математичних обчислень на мові програмування Python.

# ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вирішити завдання з цілочисельними змінними. Всі вхідні і

вихідні дані в задачах цієї групи є цілими числами. Всі числа, для яких вказано

кількість цифр (двозначне число, тризначне число та ін.), вважаються

додатними. Завдання представлено в табл.1.

Завдання 2. Обчислити математичне вираз зі змінними дійсного типу,

використовуючи стандартну бібліотеку math. Вирази представлено в табл.2.

Завдання 3. Вирішити завдання з логічними змінними. У всіх завданнях

даної групи потрібно вивести логічне значення True, якщо наведене

висловлювання для запропонованих вихідних даних є істинним, і значення

False в іншому випадку. Всі числа, для яких вказано кількість цифр (двозначне число, тризначне число і т.ін.), вважаються цілими додатними. Завдання представлено в табл.3.

ВИКОНАННЯ РОБОТИ

Завдання 1. Вирішення задачі Integer 8

Вхідні дані: Дано двозначне число. Вивести число, отримане при перестановці цифр вихідного числа.

**number** — двозначне число, введене користувачем. Тип даних: int. Обмеження: повинно бути в діапазоні від 10 до 99 включно.

Вихідні дані:

swapped\_number — число, отримане після перестановки цифр вихідного числа. Тип даних: int.

error\_message — текстове повідомлення про помилку, якщо введене число не є двозначним. Тип даних: str.

Алгоритм вирішення:

Введення даних:

Користувач вводить число, яке має бути двозначним. Це число зберігається в змінній number.

Перевірка коректності введення:

Програма перевіряє, чи входить введене число в діапазон від 10 до 99.

Якщо число не відповідає цим умовам, програма виводить повідомлення про помилку і завершує виконання.

Розбиття числа на цифри:

Щоб отримати десятки, програма ділить число націло на 10: tens=number//10tens = number // 10tens=number//10.

Щоб отримати одиниці, програма обчислює залишок від ділення на 10: ones=number%10ones = number \% 10ones=number%10.

Перестановка цифр:

Для перестановки цифр місцями, програма формує нове число: swapped\_number=ones×10+tensswapped\\_number = ones \times 10 + tensswapped\_number=ones×10+tens.

Виведення результату:

Якщо число було введене коректно, програма виводить нове число з переставленими цифрами.

Якщо введене число було некоректним, виводиться повідомлення про помилку.

Лістинг коду вирішення задачі наведено в дод.А стор. 10. Екран роботи програми показаний на рис. Б.1.

ВИСНОВКИ

Було вивчено процес роботи з двозначними числами та їх обробку в Python. На практиці закріплено навички розбиття числа на цифри, перевірки вхідних даних і перестановки цифр місцями. У коді програми відпрацьовано обробку можливих помилок та реалізацію алгоритму для виведення результату.

Завдання 2 Вирішення задачі math 12

Вхідні дані:

x — змінна, введена користувачем. Тип даних: float. Обмеження: x>2.5x > 2.5x>2.5. Використовується для обчислення математичного виразу.

numerator — чисельник виразу, що включає тангенс абсолютного значення виразу 2x2+5x−31.152x^2 + 5x - 31.152x2+5x−31.15 і логарифм за основою 5 від модуля x−2.5x - 2.5x−2.5. Тип даних: float.

denominator — знаменник виразу, який є кубічним коренем суми квадрата синуса x3x^3x3 та частини виразу, що включає логарифм за основою 5 від x2−2.5x^2 - 2.5x2−2.5. Тип даних: float.

Вихідні дані:

y — результат обчислення математичного виразу. Тип даних: float.

error\_message — текстове повідомлення про помилку, якщо введене число не відповідає умовам або виникає помилка в процесі обчислення. Тип даних: str.

Алгоритм вирішення:

Введення значення змінної:

Користувач вводить значення змінної xxx, яке має бути більше 2.5.

Програма перевіряє коректність введення (чи є введене значення числом і чи відповідає воно обмеженням). Якщо введене значення не відповідає умовам, виводиться повідомлення про помилку, і програма припиняє виконання.

Обчислення чисельника:

Обчислюється чисельник виразу: тангенс абсолютного значення виразу 2x2+5x−31.152x^2 + 5x - 31.152x2+5x−31.15 плюс логарифм за основою 5 від модуля x−2.5x - 2.5x−2.5.

Обчислення знаменника:

Обчислюється знаменник виразу: кубічний корінь з суми квадрата синуса x3x^3x3 та частини виразу, що містить логарифм за основою 5 від x2−2.5x^2 - 2.5x2−2.5.

Перевірка на коректність обчислень:

Програма перевіряє, чи не виникають помилки під час виконання математичних операцій (наприклад, якщо знаменник дорівнює нулю або виникають інші помилки при логарифмуванні чи тригонометричних функціях).

Обчислення результату:

Якщо всі обчислення пройшли успішно, програма обчислює значення yyy, яке дорівнює відношенню чисельника до знаменника.

Виведення результату:

Якщо обчислення були виконані коректно, програма виводить значення yyy.

Якщо виникає помилка на будь-якому етапі, програма виводить відповідне повідомлення про помилку і завершує робот

Лістинг коду вирішення задачі наведено в дод. А.стор 10. Екран роботи програми показаний на рис. Б.2.

ВИСНОВКИ

У результаті виконаної роботи було вивчено і застосовано на практиці обчислення складних математичних виразів за допомогою модуля `math` у Python. Було закріплено навички роботи з логарифмами, тригонометричними функціями, обробкою винятків, а також проведено перевірку на коректність введених користувачем даних. Реалізовано алгоритм для обчислення заданого виразу, що дозволило отримати правильні результати або відповідні повідомлення про помилки.

Завдання 3 Вирішення задачі Boolean 19

Вхідні дані:

Надаються три цілі числа, для яких необхідно перевірити наявність хоча б однієї пари взаємно протилежних чисел.

Тип даних**:**

a — ціле число, може бути як додатним, так і від'ємним.

b — ціле число, може бути як додатним, так і від'ємним.

c — ціле число, може бути як додатним, так і від'ємним.

Обмеження**:**

Цілі числа a, b, c належать множині цілих чисел Z, тобто можуть приймати як додатні, так і від'ємні значення або бути нулями.

Числа можуть бути різними або однаковими.

Вихідні дані:

Опис: Необхідно вивести результат у вигляді логічного значення:

True — якщо серед трьох наданих чисел є хоча б одна пара взаємно протилежних чисел (тобто таких чисел, що одне з них є протилежним значенню іншого).

False — якщо серед трьох чисел немає жодної пари взаємно протилежних чисел.

Тип даних: Логічне значення (True або False).

Алгоритм вирішення:

**Введення трьох цілих чисел**:

Програма приймає три числа a, b, c як вхідні параметри. Вхідні значення можуть бути додатними, від'ємними а бо рівними нулю.

**Перевірка взаємно протилежних чисел**:

**Крок 1**: Перевіряємо першу пару a та b: якщо a=−ba = -ba=−b, то це взаємно протилежні числа. Якщо це так, то результатом буде True.

**Крок 2**: Якщо перша пара не є протилежною, перевіряємо наступну пару a та c: якщо a=−ca = -ca=−c, то це взаємно протилежні числа. Якщо це так, то результатом буде True.

**Крок 3**: Якщо жодна з попередніх пар не є протилежною, перевіряємо третю пару b та c: якщо b=−cb = -cb=−c, то це взаємно протилежні числа. Якщо це так, то результатом буде True.

**Виведення результату**:

Якщо хоча б одна з перевірок була успішною (умови виконалися), виводимо True.

Якщо жодна з умов не виконалася, виводимо False.

Лістинг коду вирішення наведено в дод. А. стор. 10. Екран роботи програми показаний на рис. Б.3

ВИСНОВОК

Було вивчено та закріплено на практиці алгоритм перевірки взаємно протилежних чисел серед трьох цілих значень. Алгоритм було відпрацьовано в коді програми, що дозволило отримати навички обробки умовних операторів і роботи з логічними значеннями. Результати роботи демонструють ефективність перевірки чисел на взаємну протилежність.

ДОДАТОК А

Лістинг коду програми для задачі Завдання 1

def swap\_digits():

try:

# Введення двозначного числа

number = int(input("Введіть двозначне число: "))

# Перевірка, чи є введене число двозначним

if number < 10 or number > 99:

raise ValueError("Число повинно бути двозначним.")

# Отримання десятків та одиниць

tens = number // 10

ones = number % 10

# Перестановка цифр

swapped\_number = ones \* 10 + tens

# Виведення результату

print("Число після перестановки цифр:", swapped\_number)

except ValueError as e:

print(e)

# Виклик функції

swap\_digits()

Лістинг коду для Завдання 2

import math

def task2\_0(): """

Обчислення математичного виразу:

y = tg(|2 \* x^2 + 5 \* x - 31.15| + log5|x - 2.5|) /

(корінь кубічний з (sin^2(x^3) + 1/3 \* log5|x^2 - 2.5|))

"""

try:

# Введення значення змінної

x = float(input("Введіть значення x (x > 2.5): "))

if x <= 2.5:

raise ValueError("x має бути більше 2.5")

except ValueError as e:

print("Помилка:", e)

input("Натисніть Enter для виходу...")

else:

try:

# Чисельник виразу

numerator = math.tan(abs(2 \* x\*\*2 + 5 \* x - 31.15) + math.log(abs(x - 2.5), 5))

# Знаменник виразу

denominator = (math.sin(x\*\*3)\*\*2 + (1/3) \* math.log(abs(x\*\*2 - 2.5), 5)) \*\* (1/3)

# Обчислення y

y = numerator / denominator

except Exception as e:

print("Помилка в обчисленнях:", e)

input("Натисніть Enter для виходу...")

else:

print(f"Результат: y = {y}")

# Виклик функції

task2\_0()

Лістинг коду програми до Завдання 3

def check\_opposite\_pair(a, b, c):

"""

Перевіряє, чи є серед трьох цілих чисел хоча б одна пара взаємно протилежних.

Параметри:

a (int): Перше ціле число.

b (int): Друге ціле число.

c (int): Третє ціле число.

Повертає:

bool: True, якщо є хоча б одна пара взаємно протилежних чисел, інакше False.

"""

if a == -b or a == -c or b == -c:

return True

else:

return False

# Приклад використання

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

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

c = int(input("Введіть третє число: "))

result = check\_opposite\_pair(a, b, c)

print(f"Чи є хоча б одна пара взаємно протилежних чисел? {result}")

ДОДАТОК Б

Скрін-шоти вікна роботи програми

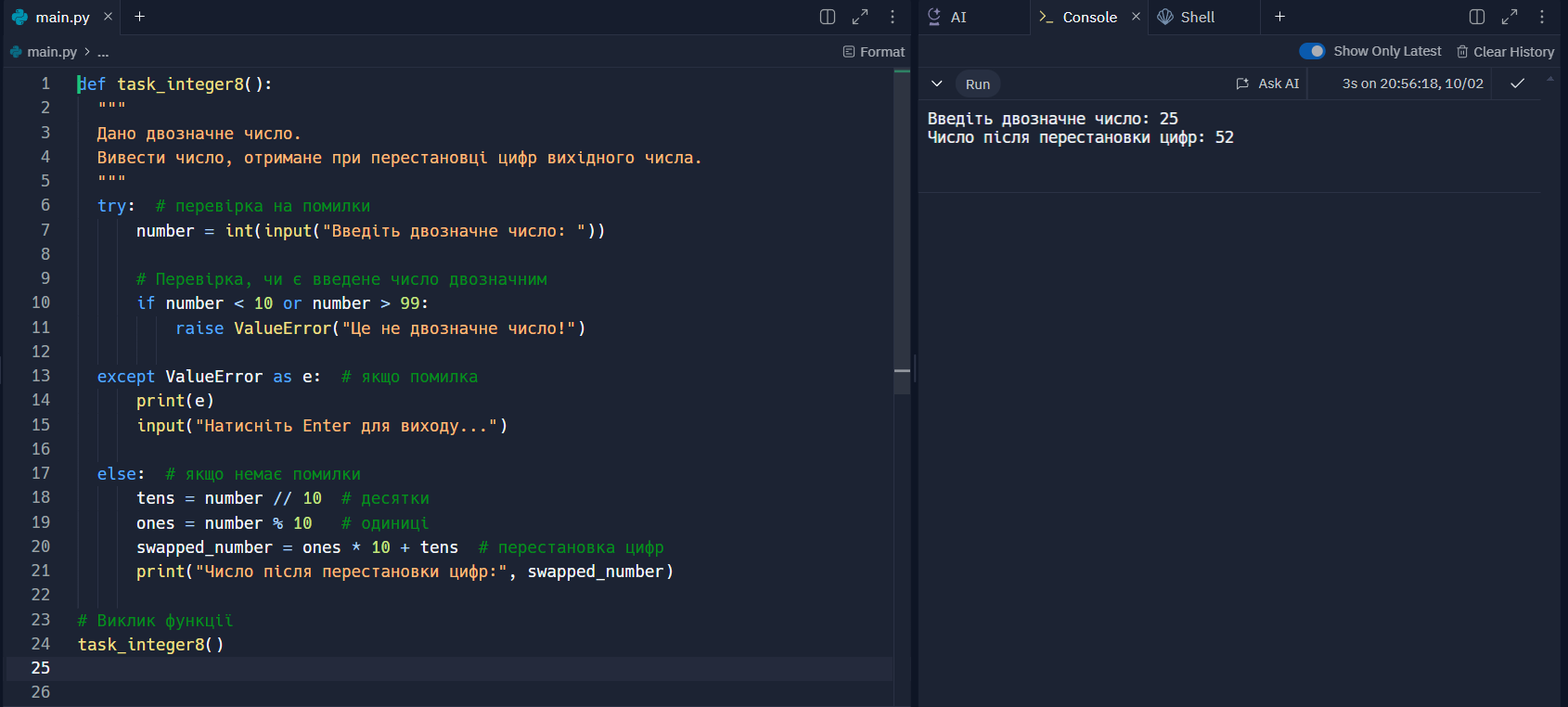
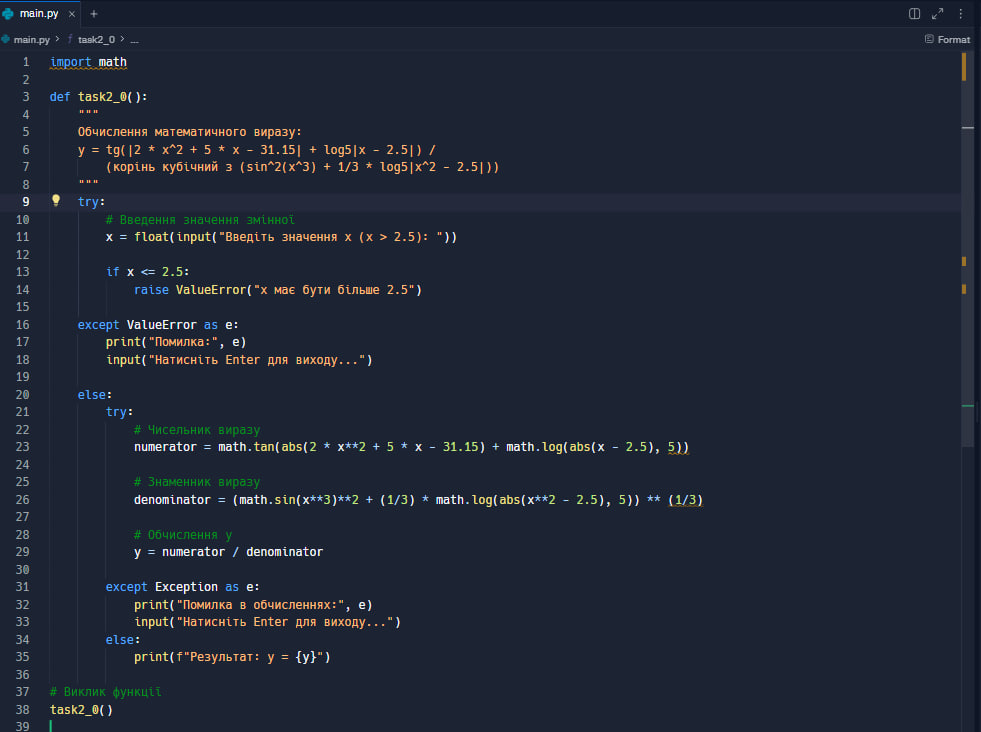


Рисунок Б.1 – Екран виконання програми для вирішення Завдання 1 (Integer)



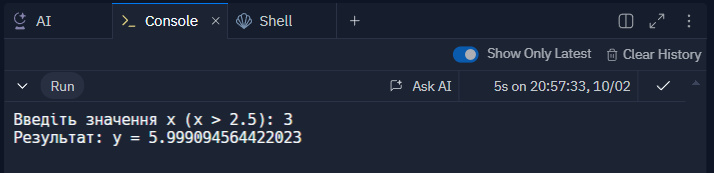


Рисунок Б.2 – Екран виконання програми для вирішення Завдання 2 (math)

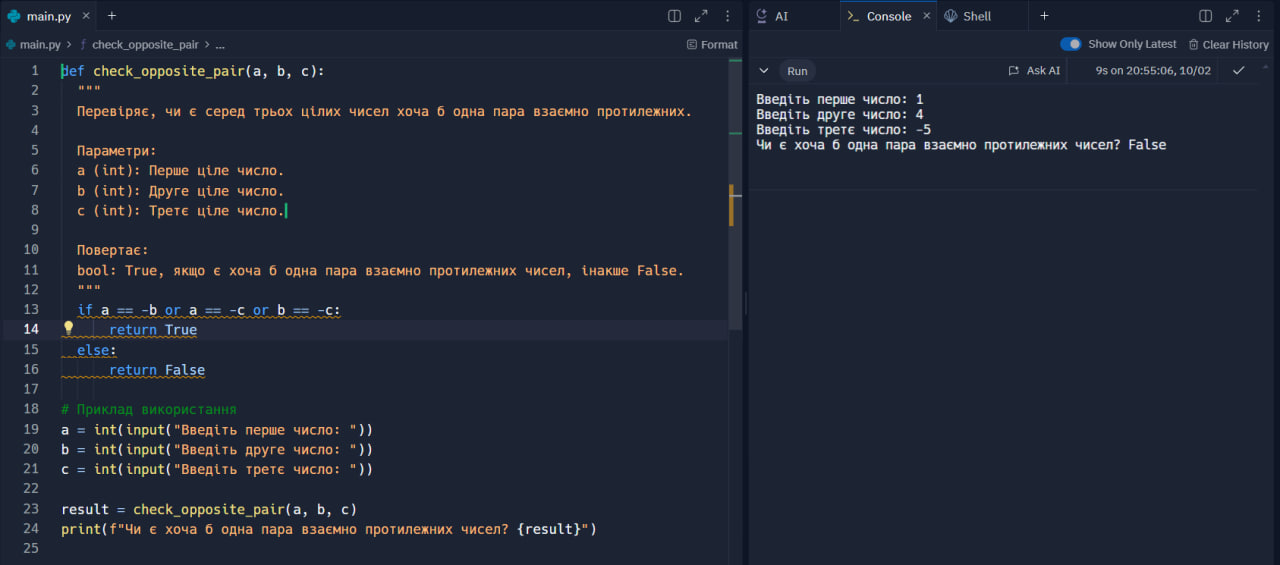


Рисунок Б.3 – Екран виконання програми для вирішення Завдання 3 (Boolean)