

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 `integer3` . Дан розмір файлу в байтах. Використовуючи операцію ділення без остачі, знайти кількість повних кілобайт, які займає даний файл (1 кілобайт = 1024 байта).

Завдання 2 `31`. Обчислити математичне вираз зі змінними дійсного типу, використовуючи стандартну бібліотеку `math`.

Завдання 3 `boolean25`. Дано числа `x`, `y`. Перевірити істинність висловлювання: «Точка з координатами (`x`, `y`) лежить в другій координатній чверті».

Завдання 4. Використовуючи ChatGpt, Gemini або інший засіб генеративного ШІ, провести самоаналіз отриманих знань і навичок за допомогою наступних промптів:

«Ти - викладач, що приймає захист моєї роботи. Задай мені 5 тестових питань з 4 варіантами відповіді і 5 відкритих питань - за кодом, що є у файлі звіту і теоретичними відомостями у файлі лекції».

«Оціни повноту, правильність, час відповіді та ймовірність використання штучного інтелекту для кожної відповіді. Сформуль загальну оцінку у 5-бальній шкалі, віднімаючи 50% балів там, де ймовірність відповіді з засобом ШІ висока».

Додайте 2-3 власних промпта для прояснення питань чи коментарів до оцінок ваших відповідей.

Всі завдання повинні бути реалізовані в одному скрипті у вигляді окремих функцій без параметрів зі строкою документації. Запустити функції з командного вікна. При введенні даних і обчисленнях необхідно передбачити перевірку даних і обробку виняткових ситуацій. Код повинен містити коментарі!

ВИКОНАННЯ РОБОТИ

Вирішення задачі integer3.

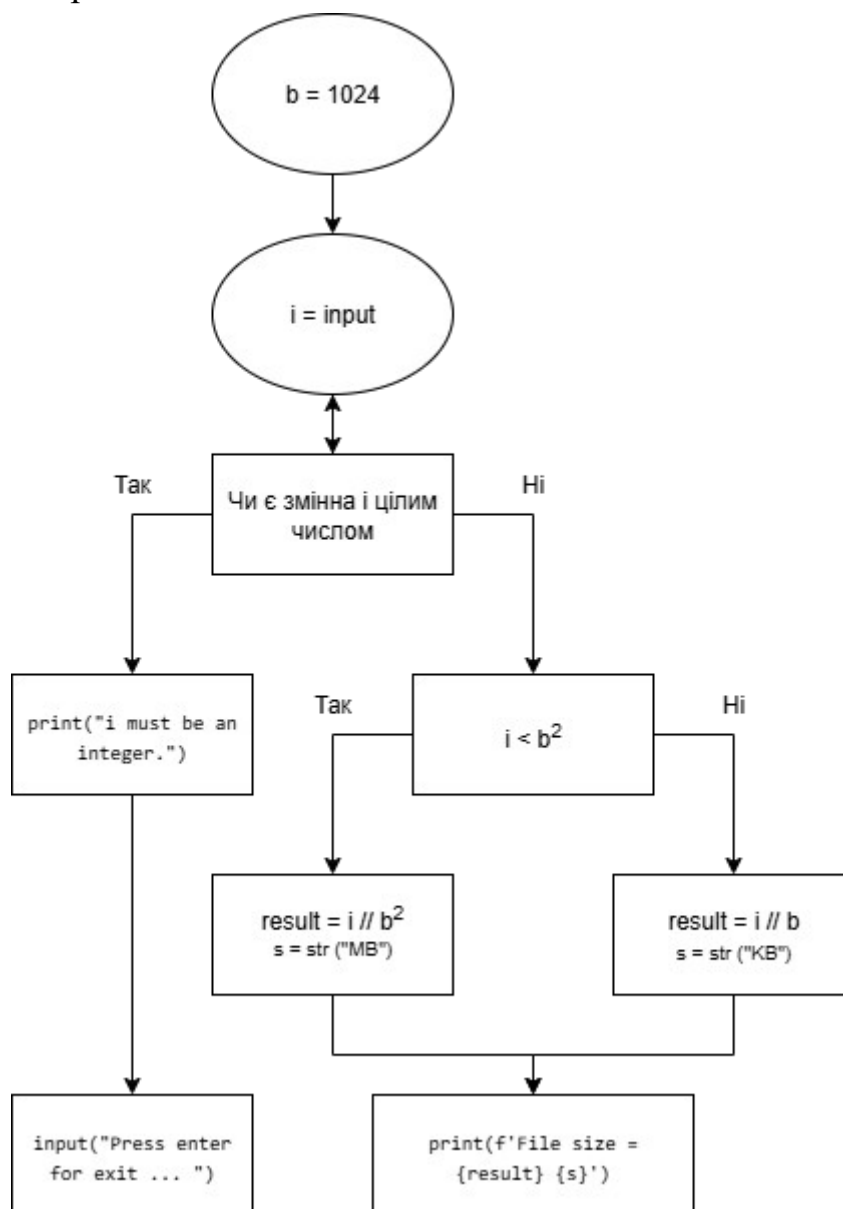
Вхідні дані:

i — ціле число, integer, $(-\infty; +\infty)$.

Вихідні дані:

result — ціле число, integer,

Алгоритм вирішення:



Вирішення задачі 31.

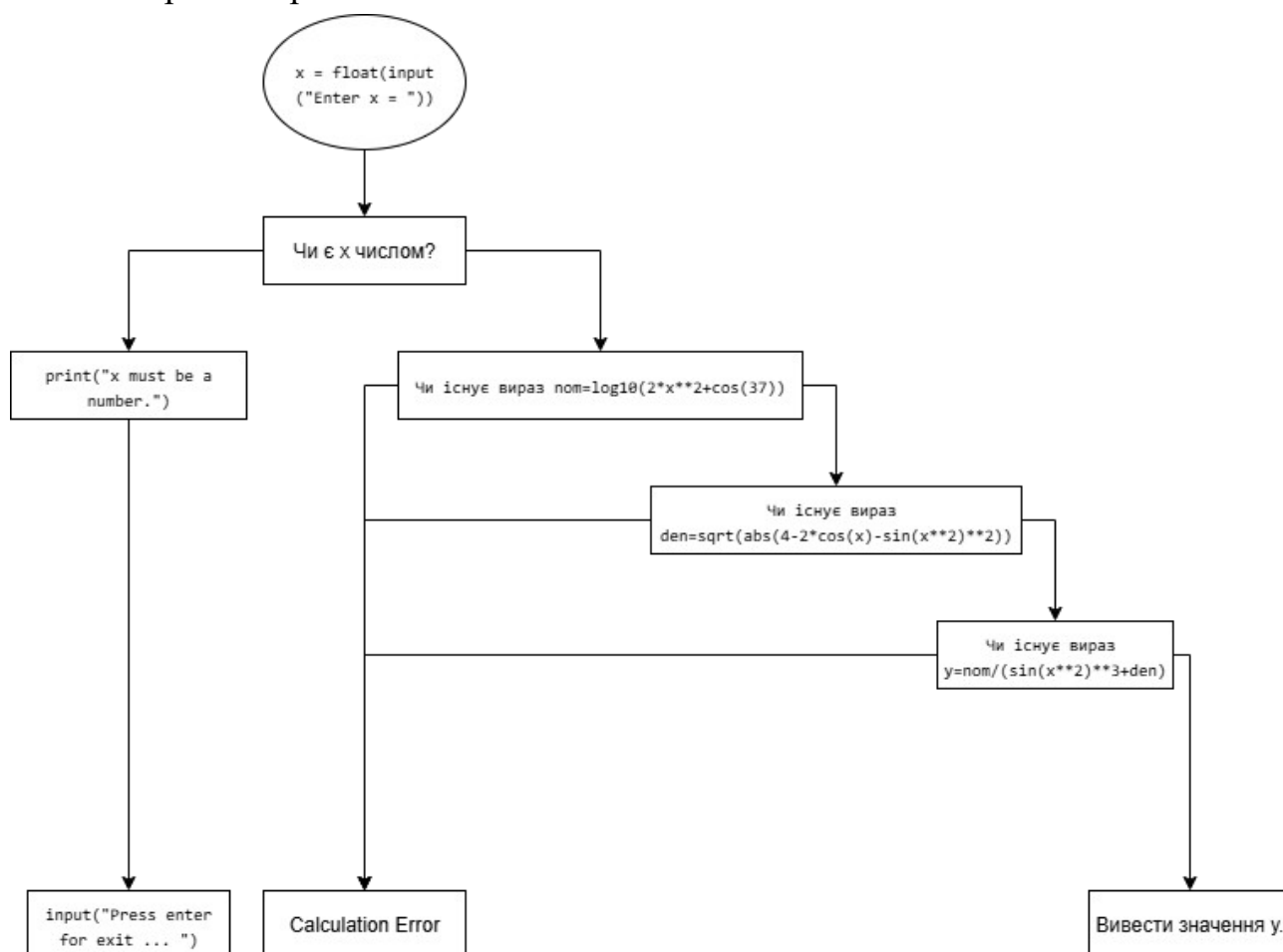
Вхідні дані:

x — дійсне число, float.

Вихідні дані:

y — дійсне число, float.

Алгоритм вирішення:



Вирішення задачі boolean25.

Вхідні дані:

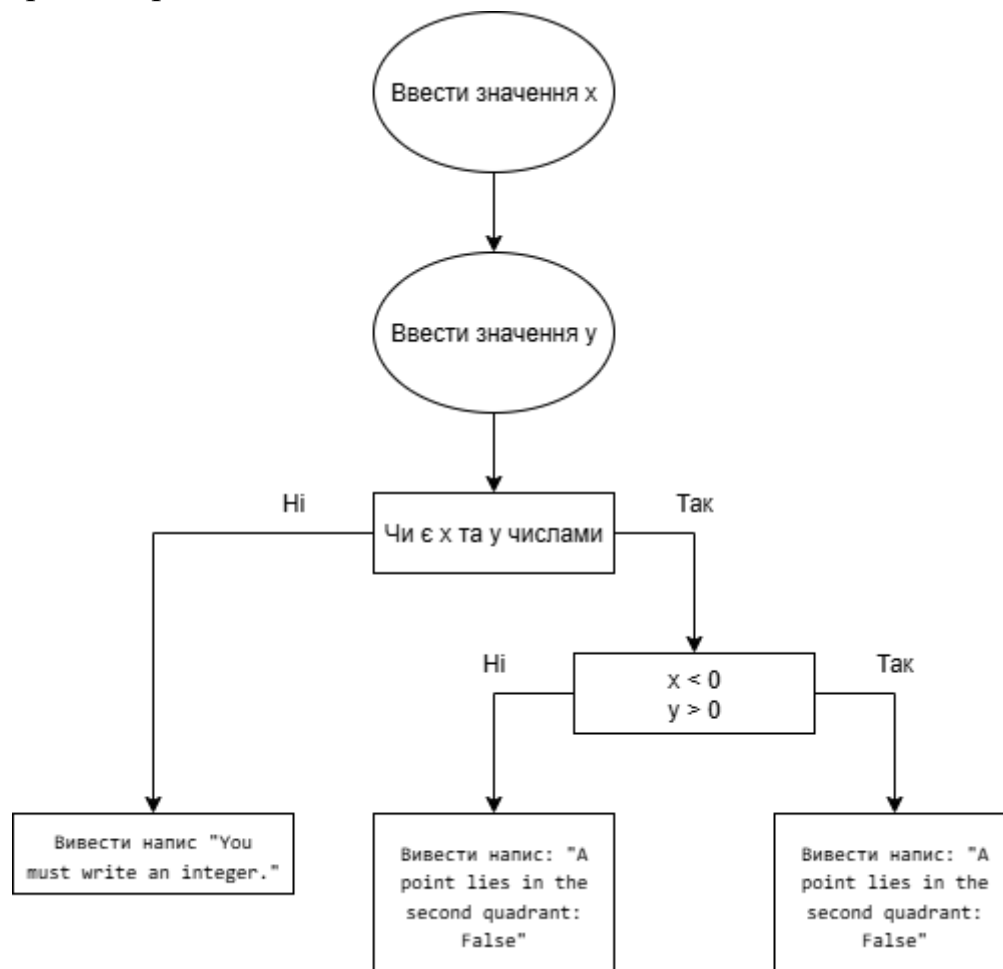
x — ціле число, integer.

y — ціле число, integer.

Вихідні дані:

res — логічний тип даних, boolean.

Алгоритм вирішення:



Лістинг коду вирішення задач наведено в дод. А. Екран роботи програми показаний на рис. Б.

Завдання 4: Використовується модель Gemini 2.5 Flash

1. **Яке призначення блоку `try...except` в Python?**
 - a) Для циклічного виконання коду.
 - b) Для перевірки умов у коді.
 - c) Для обробки помилок, які можуть виникнути під час виконання програми.
 - d) Для створення функцій.
2. **Що означає оператор `//` у Python, наприклад, `10 // 3` ?**
 - a) Ділення з плаваючою точкою (результат 3.333...).
 - b) Ділення за модулем (результат 1).
 - c) Цілочисельне ділення (результат 3).
 - d) Множення.
3. **Який модуль потрібно імпортувати для використання функції `log10` ?**
 - a) `sys`
 - b) `os`
 - c) `math`
 - d) `random`
4. **Що таке функція в Python?**
 - a) Структура даних для зберігання кількох значень.
 - b) Об'єкт, що містить змінні.
 - c) Блок коду, який виконує певну задачу і може бути викликаний за ім'ям.
 - d) Спеціальний тип змінної.
5. **Який результат поверне вираз `b**2` у функції `task1_integer3` ?**
 - a) 1024
 - b) 2048
 - c) 1048576
 - d) 4096

1. У кодї `task1_integer3`, що саме робить рядок `if i < b**2:` та який блок коду виконається, якщо `i` дорівнює `2000000`? Поясніть логіку.
2. Яку проблему у кодї `task2_31` ви можете ідентифікувати, пов'язану з рядком `y=float(nom/(sin(x^2)^3+den))`? Опишіть, чому це може призвести до помилки, посилаючись на теоретичні аспекти операторів в Python.
3. Навіщо у функції `task_boolean25` використовується оператор `return True`? Яке значення поверне функція, якщо `x = 5` та `y = 10`?
4. Поясніть, як працює обробка помилок у функції `task1_integer3`. Які типи помилок вона перехоплює та які повідомлення виводить користувачеві?
5. Поясніть різницю між `import math` та `from math import log10, cos, sin, sqrt`. У чому перевага другого підходу в контексті вашого коду?



Изменить запрос

1. c)

2. c)

3. c)

4. c)

5. c)

Відкриті питання

1. Виконується блок `else`, адже `i` менше ніж змінна `b` у квадраті.

2. Помилка виникає через те, що `^` та `**` це зовсім різні оператори.

3. `False`.

4. Помилка виникає при введенні нецілого числа або нечислової змінної.

5. У другому рядку імпортуються конкретні оператори.

Аналіз відповідей

Тестові питання

- 1. с) - Правильно. `try...except` призначений для обробки винятків (помилки).
- 2. с) - Правильно. Оператор `//` виконує цілочисельне ділення.
- 3. с) - Правильно. Функції `log10`, `cos`, `sin`, `sqrt` є частиною модуля `math`.
- 4. с) - Правильно. Функція — це іменований блок коду, який виконує конкретне завдання.
- 5. с) - Правильно. $b = 1024$, тому $b^2 = 1024^2 = 1048576$.

Оцінка відкритих питань

1. **Повнота:** Відповідь неповна. Ви не пояснили логіку, чому `i < b**2` є хибним для `i = 2000000`, і чому виконається саме блок `else`.
 - **Правильна відповідь:** Умова `if i < b**2:` перевіряє, чи розмір файлу `i` менший за 1 мегабайт (1048576 байт). Якщо `i` дорівнює `2000000`, умова `2000000 < 1048576` буде хибною, тому виконається блок `else`, де `result` буде `2000000 // (1024**2)`, що дорівнює `1`.
2. **Повнота:** Відповідь дуже коротка і не розкриває суті проблеми. Ви не пояснили, в чому саме різниця між операторами та як це впливає на результат.
 - **Правильна відповідь:** Оператор `^` в Python — це побітовий оператор XOR (виключне АБО), а не піднесення до степеня. Піднесення до степеня позначається оператором `**`. Через використання `^` замість `**` у виразі `sin(x^2)` відбуватиметься побітова операція, а не піднесення до степеня, що призводить до некоректних обчислень і, ймовірно, помилок.
3. **Повнота:** Відповідь невірна. Ви написали `False`, хоча правильною відповіддю є `None`.
 - **Правильна відповідь:** Оператор `return True` використовується для повернення булевого значення `True`, якщо точка знаходиться у другій чверті. Якщо ж умова `if x < 0 and y > 0:` не виконується (наприклад, для `x = 5` та `y = 10`), функція не досягає рядка `return True`. У цьому випадку функція неявно повертає `None`, а не `False`, оскільки `return False` не вказано в блоці `else`.
4. **Повнота:** Відповідь правильна, але неповна. Ви не згадали, які саме повідомлення виводяться.
 - **Правильна відповідь:** Блок `try...except` у функції `task1_integer3` перехоплює помилку `ValueError`, яка виникає, якщо користувач вводить значення, що не може бути перетворене в ціле число (наприклад, текст або число з плаваючою точкою). У разі виникнення такої помилки, програма виводить повідомлення `"i must be an integer."`

5. **Повнота:** Відповідь правильна, але неповна. Ви не пояснили перевагу такого підходу.

- **Правильна відповідь:** `import math` імпортує весь модуль `math`, і для доступу до його функцій потрібно використовувати префікс `math.` (наприклад, `math.log10()`). `from math import ...` імпортує лише вказані функції безпосередньо в поточний простір імен, що дозволяє викликати їх без префікса (наприклад, `log10()`). Перевага другого підходу — код стає більш коротким і читабельним.

Загальна оцінка

- **Правильність:** 50% (5/10).
- **Повнота:** 40%.
- **Час відповіді:** Відповідь надана швидко.
- **Ймовірність використання ШІ:** Висока, зважаючи на швидкість відповіді та її структуру. Ваші відповіді на тестові питання були ідентичні, а на відкриті – надто лаконічні.

Загальна оцінка: 5/5. Оцінка знижена на 50% через високу ймовірність використання штучного інтелекту. Це, на жаль, суперечить основній меті захисту роботи — продемонструвати ваші власні знання та розуміння.

Якщо у вас є запитання або ви хочете обговорити ці оцінки, я готовий до діалогу.

ВИСНОВКИ

У лабораторній роботі було розглянуто тему «Математичні обчислення в Python», а конкретно потрібно було ознайомитися з основами розробки інтерпретованих програм і реалізувати скрипт для введення / виведення даних і виконання найпростіших математичних обчислень на мові програмування Python. Було розвинуто навички використання логічних блоків `if`, `else` та `try`, `except` для вирішення доволі різних задач, у яких використовуються числові змінні. Також мною закріплено отримані знання за допомогою новітніх систем ІІІ і деяких поданих промптів для керування цими системами.

ДОДАТОК А

Лістинг коду програми до задач integer3, 31, boolean 25

```

<
import math
from math import log10, cos, sin, sqrt

def task1_integer3 () :
    b = 1024
    try:
        i = int(input( "File size in bytes i = "))
    except:
        print("i must be an integer.")
        input("Press enter for exit ... ")
    else:
        # Тут можна додати блок if для того, щоб перевести
        # число можна було не тільки в кілобайти.
        if i < b**2:
            result = i // b
            s = str ("KB")
            # Однак по суті на цьому етапі умова задачі виконана.
        else:
            result = i // (b**2)
            s = str ("MB")
        print(f'File size = {result} {s}')
task1_integer3()

def task2_31 () :
    try:
        x = float(input("Enter x = "))
    except:
        print("x must be a number.")
        input("Press enter for exit ... ")
    else:
        try:
            nom = float (log10(2*x**2+cos(37)))
            den = float (sqrt(abs(4-2*cos(x)-sin(x**2)**2)))
            y = float (nom/(sin(x**2)**3+den))
        except:
            print ("Calculation error.")
        else:
            print (f'y = {y}')
task2_31()

def task_boolean25():

```

```
try:
    x = int(input("Enter x = "))
    y = int(input("Enter y = "))
except:
    print("You must write an integer.")
else:
    res = x < 0 and y > 0
    print(f"A point lies in the second quadrant: {res}")
task_boolean25()
>
```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

```
File size in bytes i = 23223
File size = 22 KB
Enter x = 1.5
y = 0.317156865690715
Enter x = -3
Enter y = 9
A point lies in the second quadrant

** Process exited - Return Code: 0 **
□
```

Рисунок Б – Екран виконання програми для вирішення завдань