

ASSIGNMENT - 4.3

2303A51060

Batch-10

Task-1

Prompt: Give me a program to zero-short prompting to check a leap year, and give instructions without providing examples

code :

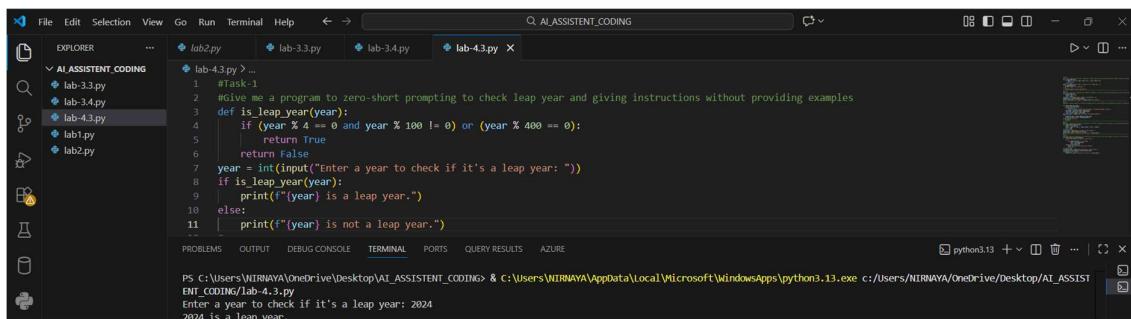
```
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    return False

year = int(input("Enter a year to check if it's a leap year: "))

if is_leap_year(year):
    print(f"{year} is a leap year.")

else:
    print(f"{year} is not a leap year.)
```

Output :



The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows files in the "AI ASSISTENT_CODING" folder: lab2.py, lab-3.3.py, lab-3.4.py, lab-4.3.py, lab1.py, and lab2.py.
- Code Editor:** The active file is lab-4.3.py, containing the provided Python code for checking leap years.
- Terminal:** The terminal window shows the command PS C:\Users\NIRNAYA\OneDrive\Desktop\AI_ASSISTENT_CODING> & C:\Users\NIRNAYA\AppData\Local\Microsoft\WindowsApps\python3.13.exe c:/Users/NIRNAYA/OneDrive/Desktop/AI_ASSISTENT_CODING/lab-4.3.py and the output Enter a year to check if it's a leap year: 2024 2024 is a leap year.

Code Analysis:

- This program determines whether a given year is a leap year using a function.
- The function applies standard leap year rules and returns True or False.
- The user inputs a year, which is checked by the function.
- The result is printed as either a leap year or not.

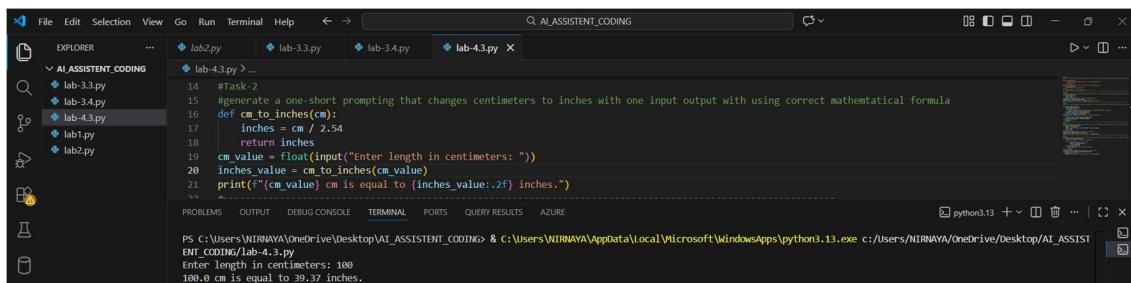
Task-2

Prompt: generate a one-short prompt that changes centimetres to inches with one input and output using the correct mathematical formula

Code :

```
def cm_to_inches(cm):  
    inches = cm / 2.54  
    return inches  
  
cm_value = float(input("Enter length in centimeters: "))  
  
inches_value = cm_to_inches(cm_value)  
  
print(f"{cm_value} cm is equal to {inches_value:.2f} inches.")
```

Output :



The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files in the "AI_ASSISTANT_CODING" folder: lab2.py, lab-3.3.py, lab-3.4.py, lab-4.3.py, lab-4.3.py (selected), lab1.py, and lab2.py.
- Code Editor:** Displays the Python code for Task-2.
- Terminal:** Shows the command "PS C:\Users\NIRNAYA\OneDrive\Desktop\AI_ASSISTANT_CODING> & C:\Users\NIRNAYA\AppData\Local\Microsoft\WindowsApps\python3.13.exe c:/Users/NIRNAYA/OneDrive/Desktop/AI_ASSISTANT_CODING\lab-4.3.py" and the user input "Enter length in centimeters: 100".
- Status Bar:** Shows "python3.13" and other status indicators.

Code Analysis:

- This program converts a length from centimetres to inches using the correct mathematical formula.
- A function performs the conversion by dividing the value by 2.54.
- The user enters a value in centimetres, which is passed to the function.
- The converted result is displayed in inches.

Task-3

Prompt: generate a python program is few-short prompting that is name formating like accepting fullname as firstname,lastname.

Code :

```
def format_name(full_name):  
    parts = full_name.split(',')  
  
    if len(parts) != 2:  
        raise ValueError("Please enter the name in 'Firstname,Lastname' format.")  
  
    first_name = parts[0].strip().capitalize()
```

```

last_name = parts[1].strip().capitalize()

return f"{first_name} {last_name}"

full_name_input = input("Enter full name (Firstname,Lastname): ")

try:

    formatted_name = format_name(full_name_input)

    print(f"Formatted Name: {formatted_name}")

except ValueError as e:

    print(e)

```

Output:

The screenshot shows the Visual Studio Code interface with the file `lab-4.3.py` open in the editor. The terminal at the bottom displays the execution of the script. It prompts the user to enter a full name, receives the input "mula,nirnaya", and then prints the formatted name "Mula Nirnaya".

```

File Edit Selection View Go Run Terminal Help ↻ 🔍 AI_ASSISTENT_CODING
EXPLORER ... lab-2.py lab-3.py lab-3.4.py lab-4.3.py
🔍 lab-3.4.py
⚙️ lab-4.3.py
⚙️ lab1.py
⚙️ lab2.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS AZURE
PS C:\Users\NIRNAYA\OneDrive\Desktop\AI_ASSISTENT_CODING> & C:\Users\NIRNAYA\AppData\Local\Microsoft\WindowsApps\python3.13.exe c:/Users/NIRNAYA/OneDrive/Desktop/AI_ASSISTENT_CODING/lab-4.3.py
Enter full name (Firstname,Lastname): mula,nirnaya
Formatted Name: Mula Nirnaya

```

Code Analysis :

- This program formats a full name entered as first name and last name.
- The input is split and validated to ensure the correct format.
- Each part of the name is cleaned and capitalised.
- The formatted full name is then displayed.

Task-4

Prompt: generate a comparative analysis for zero-short vs few-short prompting to count vowels in a string using a function

Code :

```

def count_vowels(input_string):

    vowels = 'aeiouAEIOU'

    count = sum(1 for char in input_string if char in vowels)

    return count

# Example usage

```

```

test_string = input("Enter a string to count vowels: ")

vowel_count = count_vowels(test_string)

print(f"The number of vowels in the string is: {vowel_count}")

```

Output:

The screenshot shows a code editor interface with several tabs open. The active tab is 'lab-4.3.py'. The code in the editor is:

```

40 #Task-4
41 #generate comparative analysis for zero-shot vs few-short prompting to count vowels in a string using function
42 def count_vowels(input_string):
43     vowels = "aeiouAEIOU"
44     count = sum(1 for char in input_string if char in vowels)
45     return count
46 # Example usage
47 test_string = input("Enter a string to count vowels: ")
48 vowel_count = count_vowels(test_string)
49 print(f"The number of vowels in the string is: {vowel_count}")

```

Below the code, the terminal window shows the execution of the script:

```

PS C:\Users\NIRNAYA\OneDrive\Desktop\AI_ASSISTANT_CODING & C:\Users\NIRNAYA\AppData\Local\Microsoft\WindowsApps\python3.13.exe c:/Users/NIRNAYA/OneDrive/Desktop/AI_ASSISTANT_CODING/lab-4.3.py
Enter a string to count vowels: Nirnaya
The number of vowels in the string is: 3

```

Code Analysis :

- The function counts vowels in a given string using a direct logic approach.
- Zero-shot prompting applies the logic without examples.
- Few-shot prompting helps by showing patterns before execution.
- The function returns the total number of vowels in the input string.

Task-5

Prompt: generate a few short prompts for file handling to give a read text file, count the number of lines in the file, and line count by function

```

def count_lines_in_file(file_path):

    try:

        with open(file_path, 'r') as file:

            lines = file.readlines()

            return len(lines)

    except FileNotFoundError:

        print("The specified file was not found.")

        return None

# Example usage

file_path_input = input("Enter the path of the text file: ")

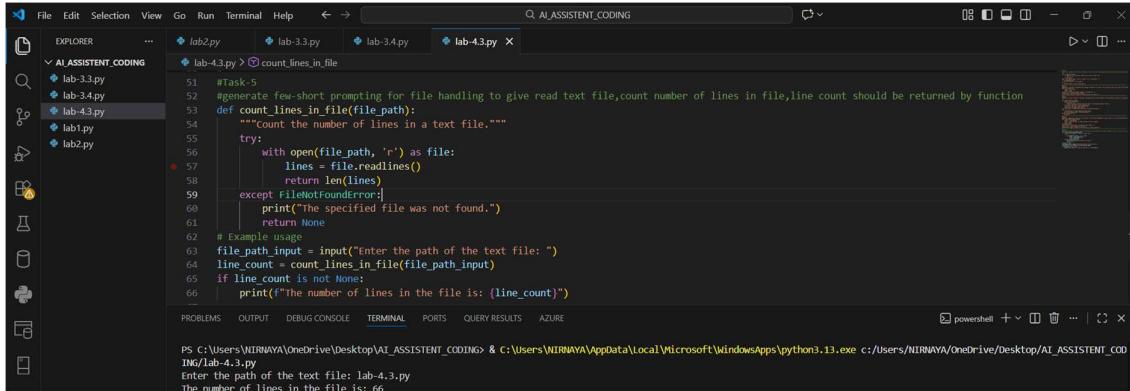
line_count = count_lines_in_file(file_path_input)

if line_count is not None:

```

```
print(f"The number of lines in the file is: {line_count}")
```

Output :



The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows files in the "AI_ASSISTANT_CODING" folder: lab2.py, lab-3.3.py, lab-3.4.py, lab-4.3.py, lab1.py, and lab2.py.
- Code Editor:** The active tab is "lab-4.3.py". The code defines a function `count_lines_in_file` that reads a text file and returns the number of lines. It includes error handling for file not found.
- Terminal:** Below the code editor, the terminal window shows the command `python lab-4.3.py` being run, followed by the output "Enter the path of the text file: lab-4.3.py" and the result "The number of lines in the file is: 66".

Code Analysis :

- This program reads a text file and counts the number of lines.
- A function opens the file safely and calculates the line count.
- Error handling is used if the file does not exist.
- The final line count is returned and displayed.