

ASSIGNMENT - 4.3

2303A51355

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.")
```

:

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

- File Explorer:** Shows files like 'AI ASSISTANT', '2303A51355_ASSIGNMENT-6.3.docx', and several Python files ('lab-1.py', 'lab-2.py', 'lab-3.py', 'lab-4.py', 'lab-5.py', 'lab-6.py', 'practice.py').
- Code Editor:** The active file is 'lab-4.py'. The code is as follows:

```
#Task-1  
#Give me a program to zero-short prompting to check leap year and giving instructions without providing examples  
  
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.")
```

- Terminal:** Shows the command run and the output:
PS C:\Users\saipr\OneDrive\Desktop\AI assistant> & 'c:\Users\saipr\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saipr\.vscode\extensions\ms-python.python.debugger-2025.18.0-win32-x64\bundled\libs\debugger\launcher' '55238' -- 'c:\Users\saipr\OneDrive\Desktop\AI assistant\lab-4.py'
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.

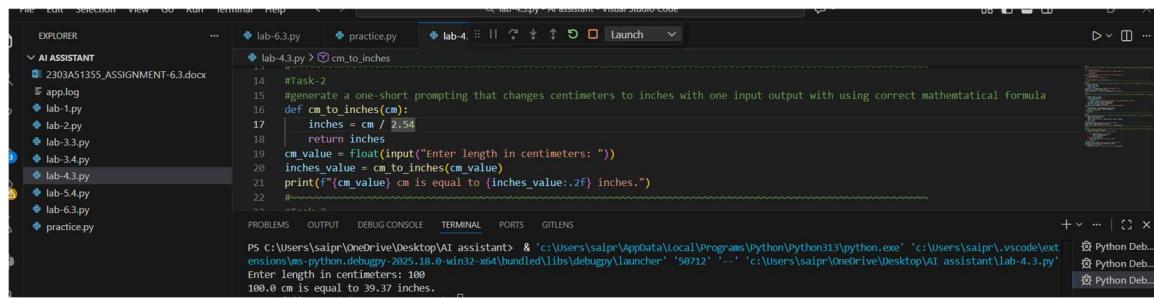
Task-2

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 :



```
#task-2
#generate a one-short prompting that changes centimeters to inches with one input output with using correct mathematical formula
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.")
```

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 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:
    forma ed_name = format_name(full_name_input)    print(f'Forma
ed      Name: {forma ed_name}')
except ValueError as e:
    print(e)  Output:

```

The screenshot shows the Visual Studio Code interface. The left sidebar displays a file tree under 'AI ASSISTANT' containing several Python files: lab-6.3.py, lab-4.3.py, lab-1.py, lab-2.py, lab-3.py, lab-3.4.py, lab-4.3.py, lab-5.4.py, lab-6.3.py, and practice.py. The main code editor window shows the content of lab-4.4.py. The terminal window at the bottom shows the command PS C:\Users\saipr\OneDrive\Desktop\AI assistant & 'c:\Users\saipr\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saipr\.vscode\extensions\ms-python.python.debug-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59725' '-d' 'c:\Users\saipr\OneDrive\Desktop\AI assistant\lab-4.3.py'. It then prompts 'Enter full name (Firstname,Lastname):' followed by 'yamsani,sai' and outputs 'Formatted Name: Yamsani Sai'. A status bar at the bottom right indicates multiple Python debugging sessions.

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}")

```

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

- File Explorer:** Shows files in the AI ASSISTANT folder: 2303A51355_ASSIGNMENT-6.3.docx, app.log, lab-1.py, lab-2.py, lab-3.py, lab-3.4.py, lab-4.3.py, lab-5.py, lab-6.3.py, and practice.py.
- Code Editor:** The active file is lab-4.3.py, containing a function to count vowels in a string.
- Terminal:** The terminal shows the command run and the output of the script execution.
- Status Bar:** Shows the Python environment and debugger status.

```

38 |     print(e)
39 | 
40 | #task-4
41 | #generate comparative analysis for zero-shot vs few-shot 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 | 
47 | # Example usage
48 | test_string = input("Enter a string to count vowels: ")
49 | vowel_count = count_vowels(test_string)
50 | print(f"The number of vowels in the string is: {vowel_count}")
51 | 
52 | #~~~~~#

```

```

PS C:\Users\saipr\OneDrive\Desktop\AI assistant> & "C:\Users\saipr\AppData\Local\Programs\Python\Python313\python.exe" 'c:\Users\saipr\vscode\extensions\ms-python.python.debug-2025.19.0-win32-x64\bundled\libs\debug\launcher' '59798' '--' 'c:\Users\saipr\OneDrive\Desktop\AI assistant\lab-4.3.py'
Enter a string to count vowels: saiprasanna
The number of vowels in the string is: 5

```

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 Visual Studio Code interface. The left sidebar displays a file tree under 'EXPLORER' with files like 'AI ASSISTANT', '2303A51355_ASSIGNMENT-6.3.docx', and several Python files including 'lab-1.py', 'lab-2.py', 'lab-3.py', 'lab-4.py', 'lab-5.py', 'lab-6.py', and 'practice.py'. The main editor area contains the provided Python code. The bottom right corner shows the 'OUTPUT' tab with the terminal output:

```

PS C:\Users\saipr\Desktop\AI assistant> & 'c:\Users\saipr\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\saipr\.vscode\extensions\ms-python.python.debug-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '61942' '--' 'c:\Users\saipr\Desktop\AI assistant\lab-4.3.py'
Enter the path of the text file: lab-4.3.py
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.