

Assingment-4.3

Task 1: Zero-Shot Prompting – Leap Year Check

Scenario

Zero-shot prompting involves giving instructions without providing examples.

Task Description

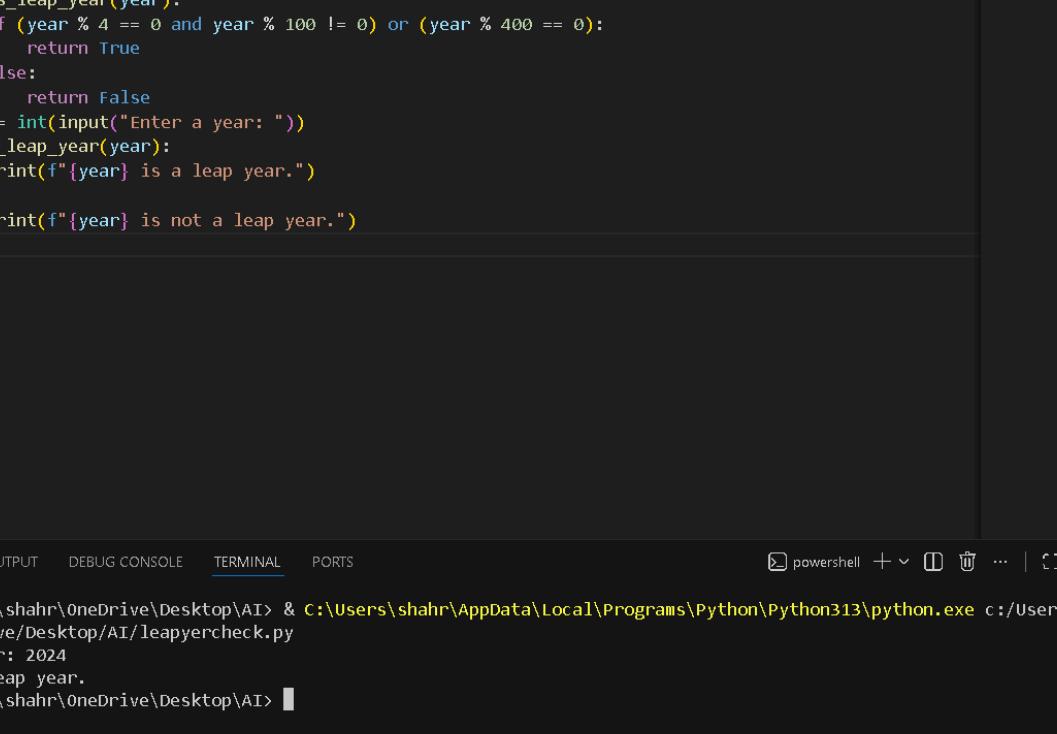
Use zero-shot prompting to instruct an AI tool to generate a Python function that:

- *Accepts a year as input*
- *Checks whether the given year is a leap year*
- *Returns an appropriate result*

Note: No input-output examples should be provided in the prompt.

Expected Output

- *AI-generated leap year checking function*
- *Correct logical conditions*
- *Sample input and output*
- *Screenshot of AI-generated response (if required)*



```
leapyearcheck.py X cmtoinches.py name.py vowels.py textfile.py perfect number.py samp Run Python File
leapyearcheck.py > ...
1 #write a code that accepts a year and to check if a year is a leap year or not user input function
2 def is_leap_year(year):
3     if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
4         return True
5     else:
6         return False
7 year = int(input("Enter a year: "))
8 if is_leap_year(year):
9     print(f"{year} is a leap year.")
10 else:
11     print(f"{year} is not a leap year.")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ⌂ ⌂ ... | ⌂ >
PS C:\Users\shahr\OneDrive\Desktop\AI> & c:\Users\shahr\AppData\Local\Programs\Python\Python313\python.exe c:/Users/shahr/OneDrive/Desktop/AI/leapyearcheck.py
Enter a year: 2024
2024 is a leap year.
PS C:\Users\shahr\OneDrive\Desktop\AI>
```

Task 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

Use one-shot prompting by providing one input-output example to generate a Python function that:

- Converts centimeters to inches
 - Uses the correct mathematical formula

Example provided in prompt:

Input: 10 cm → Output: 3.94 inches

Expected Output

- Python function with correct conversion logic
 - Accurate calculation

- Sample test cases and outputs

The screenshot shows a code editor interface with several tabs at the top: leapyearcheck.py, cmtoinches.py (which is open), name.py, vowels.py, textfile.py, perfect number.py, samp, and Below the tabs, the code for cmtoinches.py is displayed:

```

1  #write a code to convert centimeters to inches using math function user input
2  def cm_to_inches(cm):
3      inches = cm / 2.54
4      return inches
5  cm = float(input("Enter length in centimeters: "))
6  inches = cm_to_inches(cm)
7  print(f"{cm} cm is equal to {inches} inches.")

```

At the bottom of the editor, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following command-line interaction:

```

PS C:\Users\shahr\OneDrive\Desktop\AI> & C:\Users\shahr\AppData\Local\Programs\Python\Python313\python.exe c:/Users/shahr/OneDrive/Desktop/AI/cmtoinches.py
Enter length in centimeters: 562
562.0 cm is equal to 221.25984251968504 inches.
PS C:\Users\shahr\OneDrive\Desktop\AI>

```

Task 3: Few-Shot Prompting – Name Formatting

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

Use few-shot prompting with 2–3 examples to generate a Python function that:

- Accepts a full name as input
- Formats it as “Last, First”

Example formats:

- "John Smith" → "Smith, John"
- "Anita Rao" → "Rao, Anita"

Expected Output

The screenshot shows a dark-themed instance of Visual Studio Code. In the center, there's a code editor window titled 'ai.py'. The code is a Python function named 'split_full_name' that takes a full name as input and prints the first and last names separately. Below the code editor is a terminal window showing the execution of the script and its output. The terminal tabs at the bottom are labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS', with 'TERMINAL' being the active tab.

```
C: > Users > HP > Downloads > ai.py > ...
1  #write a code that accepts full name and prints first and last name separately user input function
2  def split_full_name():
3      full_name = input("Please enter your full name: ")
4      names = full_name.split()
5
6      if len(names) >= 2:
7          first_name = names[0]
8          last_name = names[-1]
9          print(f"First Name: {first_name}")
10         print(f"Last Name: {last_name}")
11     else:
12         print("Please enter both first and last names.")
13 split_full_name()
14 
```

```
PS C:\Users\HP> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/HP/Downloads/ai.py
Please enter your full name: ram sita
First Name: ram
Last Name: sita
PS C:\Users\HP>
```

- *Well-structured Python function*
- *Output strictly following example patterns*
- *Correct handling of names*
- *Sample inputs and outputs*

Task 4: Comparative Analysis – Zero-Shot vs Few-Shot

Scenario

Different prompt strategies may produce different code quality.

Task Description

- *Use zero-shot prompting to generate a function that counts vowels in a string*
- *Use few-shot prompting for the same problem*
- *Compare both outputs based on:*
 - *Accuracy*
 - *Readability*

o Logical clarity

Expected Output

- *Two vowel-counting functions*
- *Comparison table or short reflection paragraph*
- *Conclusion on prompt effectiveness*

The screenshot shows the Visual Studio Code interface. The left sidebar has icons for file, search, and other tools. The main area shows a Python script named 'ai.py' with the following code:

```
C: > Users > HP > Downloads > ai.py > ...
1 #write a code that counts vowels in a given string user input function
2 def count_vowels(input_string):
3     vowels = "aeiouAEIOU"
4     count = 0
5     for char in input_string:
6         if char in vowels:
7             count += 1
8     return count
9 # Example usage
10 user_input = input("Enter a string: ")
11 vowel_count = count_vowels(user_input)
12 print(f"The number of vowels in the given string is: {vowel_count}")
```

Below the code editor, the terminal tab is selected. The terminal window shows the command 'PS C:\Users\HP> & C:/Users/HP/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/HP/Downloads/ai.py' followed by the output 'Enter a string: assistant coding'. The terminal then prints 'The number of vowels in the given string is: 5'.

Task 5: Few-Shot Prompting – File Handling

Scenario

File processing requires clear logical understanding.

Task Description

Use few-shot prompting to generate a Python function that:

- *Reads a .txt file*
- *Counts the number of lines in the file*
- *Returns the line count*

Expected Output

- Working Python file-processing function
 - Correct line count
 - Sample .txt input and output
 - AI-assisted logic explanation