

Name: B. Srikar

Batch: 02

Course: AI ASSISSTED CODING

Semester: Even

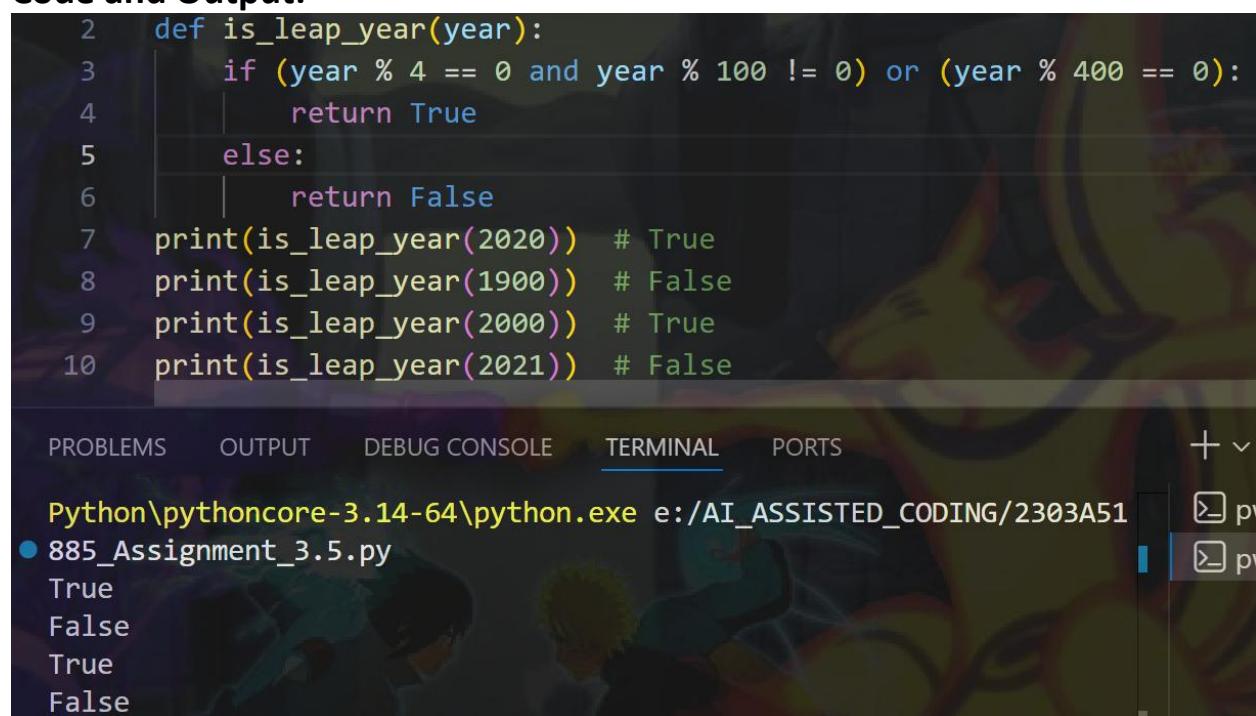
Question 1: Zero-Shot Prompting (Leap Year Check)

Write a zero-shot prompt to generate a Python function that checks whether a given year is a leap year.

Prompt:

```
#write a code to check whether the given year is leap year or not
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return True
    else:
        return False
```

Code and Output:



The screenshot shows a code editor interface with a dark theme. On the left, there is a code editor pane containing the following Python code:

```
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 print(is_leap_year(2020)) # True
8 print(is_leap_year(1900)) # False
9 print(is_leap_year(2000)) # True
10 print(is_leap_year(2021)) # False
```

Below the code editor, there is a terminal window showing the execution of the script and its output. The terminal tabs at the bottom are PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The output shows the following:

```
Python\pythoncore-3.14-64\python.exe e:/AI_ASSISTED_CODING/2303A51
● 885_Assignment_3.5.py
True
False
True
False
```

Question 2: One-Shot Prompting (GCD of Two Numbers)

Write a one-shot prompt with one example to generate a Python function that finds the Greatest Common Divisor (GCD) of two numbers.

Prompt:

"""Input: 12, 18 → Output: 6"""

```
"""
Input: 12, 18 → Output: 6
"""

def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
```

Code and Output:

```
13     Input: 12, 18 → Output: 6
14     """
● 15     def gcd(a, b):
16         while b:
17             a, b = b, a % b
18         return a
19     print(gcd(12, 18))  # Output: 6
20     print(gcd(56, 98)) # Output: 14
21     print(gcd(101, 10)) # Output: 1
22     print(gcd(48, 180)) # Output: 12
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PO

6
14
1
12

Comparing Zero Shot and One Shot Prompt:

Zero-shot prompting involves providing the model with a direct question or task without any examples. Few-shot prompting, on the other hand, includes a few examples to guide the model in understanding the task. It can improve performance on complex tasks.

Question 3: Few-Shot Prompting (LCM Calculation)

Write a few-shot prompt with multiple examples to generate a Python function that computes the Least Common Multiple (LCM).

Prompt:

```
"""• Input: 4, 6 → Output: 12  
• Input: 5, 10 → Output: 10  
• Input: 7, 3 → Output: 21 """
```

- Input: 4, 6 → Output: 12
- Input: 5, 10 → Output: 10
- Input: 7, 3 → Output: 21

```
def lcm(a, b):  
    return abs(a * b) // gcd(a, b)
```

Code and Output:

```
29  def lcm(a, b):  
30      return abs(a * b) // gcd(a, b)  
31  print(lcm(4, 6))  # Output: 12  
32  print(lcm(5, 10)) # Output: 10  
33  print(lcm(7, 3)) # Output: 21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS


```
12  
10  
21
```

Question 4: Zero-Shot Prompting (Binary to Decimal Conversion)

Write a zero-shot prompt to generate a Python function that converts a binary number to decimal.

Prompt:

#Write a zero-shot prompt to generate a Python function that converts a binary number to decimal.

```
#Write a zero-shot prompt to generate a Python function that converts
def binary_to_decimal(binary_str):
    decimal_value = 0
    binary_str = binary_str[::-1] # Reverse the string for easier cal
    for index, digit in enumerate(binary_str):
        if digit == '1':
```

Code and Output:

```
36 def binary_to_decimal(binary_str):
37     decimal_value = 0
38     binary_str = binary_str[::-1] # Reverse the string for easie
39     for index, digit in enumerate(binary_str):
40         if digit == '1':
41             decimal_value += 2 ** index
42     return decimal_value
43 print(binary_to_decimal('1010')) # Output: 10
44 print(binary_to_decimal('1111')) # Output: 15
45 print(binary_to_decimal('1001')) # Output: 9
46 print(binary_to_decimal('0001')) # Output: 1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

10
15
9
1

Question 5: One-Shot Prompting (Decimal to Binary Conversion)

Write a one-shot prompt with an example to generate a Python function that converts a decimal number to binary.

Prompt:

"""Input: 10 → Output: 1010"""

```
"""
Input: 10 → Output: 1010
"""

def decimal_to_binary(n):
    if n == 0:
        return '0'
    binary_str = ''
    while n > 0:
        binary_str = str(n % 2) + binary_str
        n = n // 2
```

Code and Output:

```
51 def decimal_to_binary(n):
52     if n == 0:
53         return '0'
54     binary_str = ''
55     while n > 0:
56         binary_str = str(n % 2) + binary_str
57         n = n // 2
58     return binary_str
59 print(decimal_to_binary(10)) # Output: 1010
60 print(decimal_to_binary(15)) # Output: 1111
61 print(decimal_to_binary(9)) # Output: 1001
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

1010

1111

1001

Question 6: Few-Shot Prompting (Harshad Number Check)

Write a few-shot prompt to generate a Python function that checks whether a number is a Harshad (Niven) number.

Prompt:

- ```
"""• Input: 18 → Output: Harshad Number
• Input: 21 → Output: Harshad Number
• Input: 19 → Output: Not a Harshad Number"""
```

```
"""
• Input: 18 → Output: Harshad Number
• Input: 21 → Output: Harshad Number
• Input: 19 → Output: Not a Harshad Number
"""

def is_harshad_number(n):
 digit_sum = sum(int(digit) for digit in str(n))
 if n % digit_sum == 0:
 return "Harshad Number"
 else: return "Not a Harshad Number"
```

### Code and Output:

```
68 def is_harshad_number(n):
69 digit_sum = sum(int(digit) for digit in str(n))
70 if n % digit_sum == 0:
71 return "Harshad Number"
72 else:
73 return "Not a Harshad Number"
• 74 print(is_harshad_number(18)) # Output: Harshad Number
75 print(is_harshad_number(21)) # Output: Harshad Number
76 print(is_harshad_number(19)) # Output: Not a Harshad Number
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

Harshad Number  
Harshad Number  
Not a Harshad Number