

## AI Assisted Coding

### LAB ASSIGNMENT – 3.5

Name : G.Abhiram

HT.No : 2303A51087

Batch :02

- 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 python function that checks whether a given year is a leap year

**Code :**

```
def is_leap_year(year):  
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
        return f"{year} is a leap year."  
    else:  
        return f"{year} is not a leap year."  
  
# Test the function  
  
try:  
    year = int(input("Enter a year: "))  
    result = is_leap_year(year)  
    print(result)  
  
except ValueError:  
    print("Invalid input. Please enter a valid year.")
```

```
AI-AC >(assign3.5.py) ...  
1 # write a python function that checks whether a given year is a leap year  
2 def is_leap_year(year):  
3     if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
4         return f"{year} is a leap year."  
5     else:  
6         return f"{year} is not a leap year."  
7 # Test the function  
8 try:  
9     year = int(input("Enter a year: "))  
10    result = is_leap_year(year)  
11    print(result)  
12 except ValueError:  
13     print("Invalid input. Please enter a valid year.")  
14
```

## Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppData/ents/CSE/3-2/AI-AC/assign3.5.py  
Enter a year: 1900  
1900 is not a leap year.  
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppData/ents/CSE/3-2/AI-AC/assign3.5.py  
Enter a year: 2000  
2000 is a leap year.  
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/AppData/ents/CSE/3-2/AI-AC/assign3.5.py  
Enter a year: 2024  
2024 is a leap year.
```

- 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 :**

a=12  
b=18  
display 6 as output

**Code:**

```
def compute_gcd(a, b):  
    while b:  
        a, b = b, a % b  
    return a  
  
# Test the function  
  
try:  
    a = int(input("Enter first positive integer: "))  
    b = int(input("Enter second positive integer: "))  
    if a <= 0 or b <= 0:  
        print("Please enter positive integers only.")  
    else:  
        result = compute_gcd(a, b)  
        print(f"The GCD of {a} and {b} is {result}.")  
except ValueError:  
    print("Invalid input. Please enter integers.")
```

The screenshot shows a code editor window with four tabs at the top: 'LAB-03-19.py', 'assign03.py', 'assign3.5.py' (which is the active tab), and 'arr'. The code in the editor is as follows:

```
AI-AC > assign3.5.py > ...
14
15     ...
16     a=12
17     b=18
18     display 6 as output
19     ...
20     def compute_gcd(a, b):
21         while b:
22             a, b = b, a % b
23         return a
24     # Test the function
25     try:
26         a = int(input("Enter first positive integer: "))
27         b = int(input("Enter second positive integer: "))
28         if a <= 0 or b <= 0:
29             print("Please enter positive integers only.")
30         else:
31             result = compute_gcd(a, b)
32             print(f"The GCD of {a} and {b} is {result}.")
33     except ValueError:
34         print("Invalid input. Please enter integers.")
35
```

**Output :**

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

4/python.exe c:/Users/abhir/OneDrive/Pictures/Documents/CSE/3-2/AI-AC/assign3.5.py
Enter first positive integer: 12
Enter second positive integer: 18
The GCD of 12 and 18 is 6.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-> &
C:/Users/abhir/AppData/Local/Programs/Python/Python314
/python.exe c:/Users/abhir/OneDrive/Pictures/Documents/
CSE/3-2/AI-AC/assign3.5.py
Enter first positive integer: 20
Enter second positive integer: 40
The GCD of 20 and 40 is 20.
```

- 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 :**

```
a=4
b=6
display 12 as output
a=5
b=10
display 10 as output
a=7
b=3
display 21 as output
```

**Code:**

```
def compute_lcm(a, b):
```

```
def gcd(x, y):
    while y:
        x, y = y, x % y
    return x

return abs(a * b) // gcd(a, b)

# Test the function

try:
    a = int(input("Enter first positive integer: "))
    b = int(input("Enter second positive integer: "))
    if a <= 0 or b <= 0:
        print("Please enter positive integers only.")
    else:
        result = compute_lcm(a, b)
        print(f"The LCM of {a} and {b} is {result}.")
except ValueError:
    print("Invalid input. Please enter integers.")
```

LAB-03-19.py assign03.py assign3.5.py X armstrong.py

AI-AC > assign3.5.py > ...

```
37     ...
38     a=4
39     b=6
40     display 12 as output
41     a=5
42     b=10
43     display 10 as output
44     a=7
45     b=3
46     display 21 as output ...
47
48 def compute_lcm(a, b):
49     def gcd(x, y):
50         while y:
51             x, y = y, x % y
52         return x
53     return abs(a * b) // gcd(a, b)
54 # Test the function
55 try:
56     a = int(input("Enter first positive integer: "))
57     b = int(input("Enter second positive integer: "))
58     if a <= 0 or b <= 0:
59         print("Please enter positive integers only.")
60     else:
61         result = compute_lcm(a, b)
62         print(f"The LCM of {a} and {b} is {result}.")
63 except ValueError:
64     print("Invalid input. Please enter integers.")
```

**Output:**

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

```
CSE/3-2/AI-AC/assign3.5.py
Enter first positive integer: 4
Enter second positive integer: 6
The LCM of 4 and 6 is 12.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-> &
C:/Users/abhir/AppData/Local/Programs/Python/Python314
/python.exe c:/Users/abhir/OneDrive/Pictures/Documents/
CSE/3-2/AI-AC/assign3.5.py
Enter first positive integer: 7
Enter second positive integer: 3
The LCM of 7 and 3 is 21.
```

- 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 python function to convert a binary number to decimal

**Code :**

```
def binary_to_decimal(binary_str):
    try:
        decimal_number = int(binary_str, 2)
        return f"The decimal equivalent of binary {binary_str} is
{decimal_number}."
    except ValueError:
        return "Invalid input. Please enter a valid binary number."
# Test the function
binary_str = input("Enter a binary number: ")
result = binary_to_decimal(binary_str)
```

```
print(result)
```

```
AI-AC > assign3.5.py > ...
65 ...
66 ...
67 write a python function to convert a binary number to decimal
68 ...
69 ...
70 def binary_to_decimal(binary_str):
71     try:
72         decimal_number = int(binary_str, 2)
73         return f"The decimal equivalent of binary {binary_str} is {decimal_number}."
74     except ValueError:
75         return "Invalid input. Please enter a valid binary number."
76 # Test the function
77 binary_str = input("Enter a binary number: ")
78 result = binary_to_decimal(binary_str)
79 print(result)
```

**Output :**

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

nts/CSE/3-2/AI-AC/assign3.5.py
Enter a binary number: 0111
The decimal equivalent of binary 0111 is 7.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/Downloads/CSE/3-2/AI-AC/assign3.5.py
Enter a binary number: 001110
The decimal equivalent of binary 001110 is 14.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:/Users/abhir/Downloads/CSE/3-2/AI-AC/assign3.5.py
Enter a binary number: 1111
The decimal equivalent of binary 1111 is 15.
```

- 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 :**

num=10  
display 1010 as output

**Code :**

```
def decimal_to_binary(num):  
    if num < 0:  
        return "Please enter a non-negative integer."  
    binary_str = bin(num).replace("0b", "")  
    return f"The binary equivalent of decimal {num} is {binary_str}."  
  
# Test the function  
try:  
    num = int(input("Enter a non-negative integer: "))  
    result = decimal_to_binary(num)  
    print(result)  
except ValueError:  
    print("Invalid input. Please enter an integer.")
```

```
AI-AC > assign3.5.py > ...
81     ...
82     num=10
83     display 1010 as output
84     ...
85     def decimal_to_binary(num):
86         if num < 0:
87             return "Please enter a non-negative integer."
88         binary_str = bin(num).replace("0b", "")
89         return f"The binary equivalent of decimal {num} is {binary_str}."
90     # Test the function
91     try:
92         num = int(input("Enter a non-negative integer: "))
93         result = decimal_to_binary(num)
94         print(result)
95     except ValueError:
96         print("Invalid input. Please enter an integer.")
```

## Output :

```
PROBLEMS      OUTPUT      DEBUG CONSOLE      TERMINAL      PORTS

ents/CSE/3-2/AI-AC/assign3.5.py
Enter a non-negative integer: 22
The binary equivalent of decimal 22 is 10110.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C
ents/CSE/3-2/AI-AC/assign3.5.py
Enter a non-negative integer: 24
The binary equivalent of decimal 24 is 11000.
PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C
ents/CSE/3-2/AI-AC/assign3.5.py
Enter a non-negative integer: 18
The binary equivalent of decimal 18 is 10010.
```

- 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 :**

```
number=18  
display harshad number as output  
number=19  
display not a harshad number as output  
number=21  
display harshad number as output
```

**Code :**

```
def is_harshad_number(num):  
    if num <= 0:  
        return "Please enter a positive integer."  
    digit_sum = sum(int(digit) for digit in str(num))  
    if num % digit_sum == 0:  
        return f"{num} is a Harshad number."  
    else:  
        return f"{num} is not a Harshad number."  
  
# Test the function  
try:  
    num = int(input("Enter a positive integer: "))  
    result = is_harshad_number(num)  
    print(result)  
except ValueError:  
    print("Invalid input. Please enter an integer.")
```

```
AI-AC > assign3.5.py > ...
99     ...
100    number=18
101    display harshad number as output
102    number=19
103    display not a harshad number as output
104    number=21
105    display harshad number as output
106    ...
107    def is_harshad_number(num):
108        if num <= 0:
109            return "Please enter a positive integer."
110        digit_sum = sum(int(digit) for digit in str(num))
111        if num % digit_sum == 0:
112            return f"{num} is a Harshad number."
113        else:
114            return f"{num} is not a Harshad number."
115    # Test the function
116    try:
117        num = int(input("Enter a positive integer: "))
118        result = is_harshad_number(num)
119        print(result)
120    except ValueError:
121        print("Invalid input. Please enter an integer.")
```

## Output :

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL	PORTS
ents/CSE/3-2/AI-AC/assign3.5.py				
	Enter a positive integer: 18			
	18 is a Harshad number.			
	PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2>			
ents/CSE/3-2/AI-AC/assign3.5.py				
	Enter a positive integer: 24			
	24 is a Harshad number.			
	PS C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2> & C:\Users\abhir\OneDrive\Pictures\Documents\CSE\3-2>			
ents/CSE/3-2/AI-AC/assign3.5.py				
	Enter a positive integer: 19			
	19 is not a Harshad number.			