AI Assissted Coding

LAB ASSIGNMENT – 3.5

Name : G.Abhiram

HT.No : 2303A51087

Batch :02

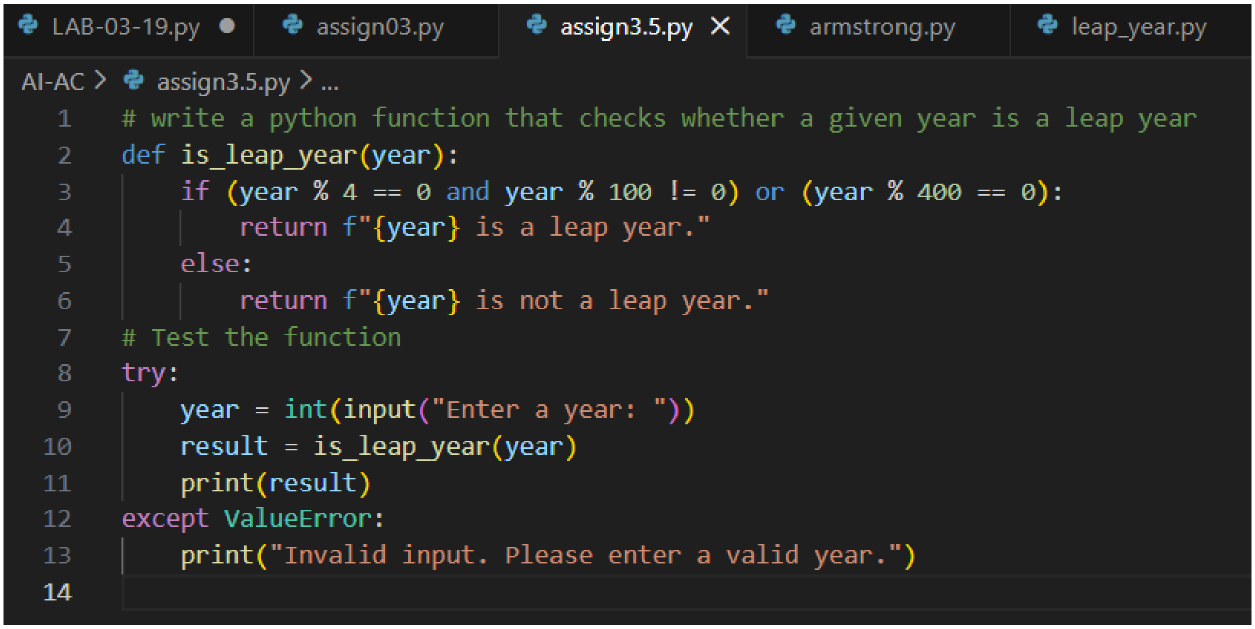
1) **Zero-Shot Prompting (Leap Year Check) . Write a zero-shot prompt to generate a Python function that checks .whether a given year is a leap year.**

Prompt :

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

Code :

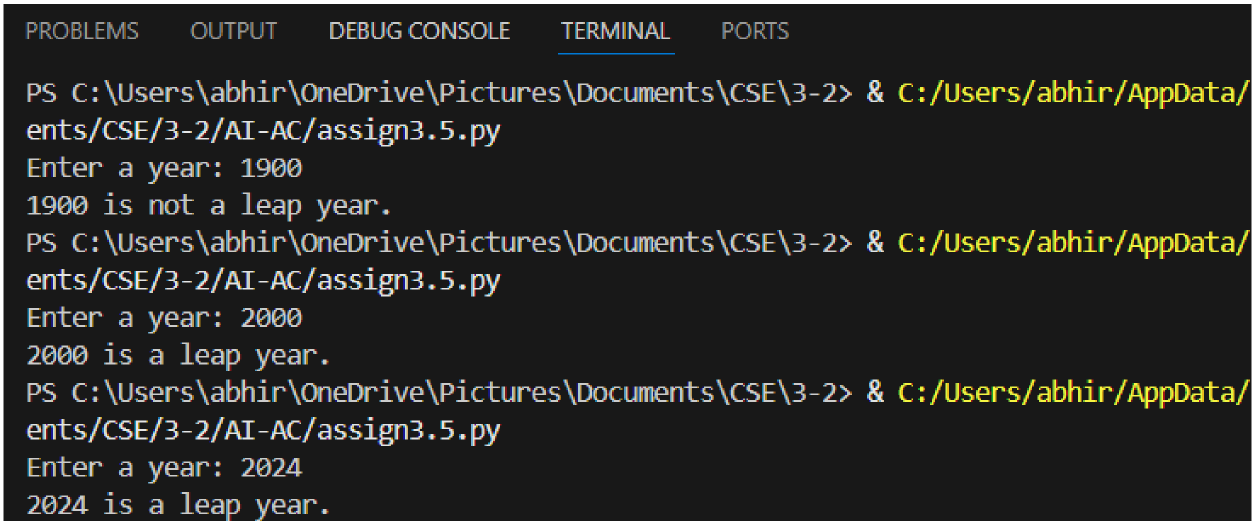
```
def is_leap_year(year):
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
        return f"{year} is a leap year."
    else:
        return f"{year} is not a leap year."
# Test the function
try:
    year = int(input("Enter a year: "))
    result = is_leap_year(year)
    print(result)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



```python
# write a python function that checks whether a given year is a leap year
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.")

```

Output :



```
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 :

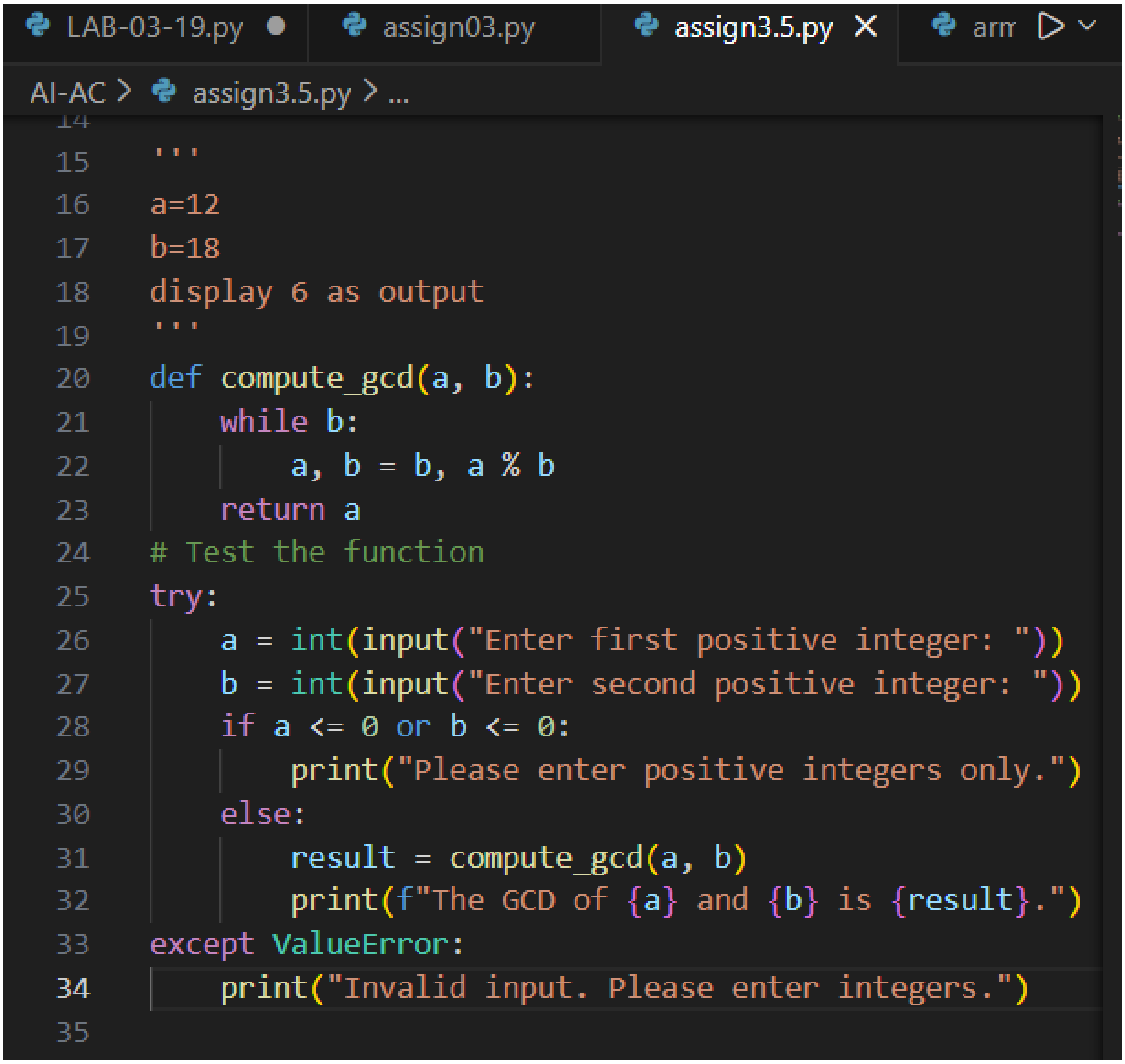
a=12

b=18

display 6 as output

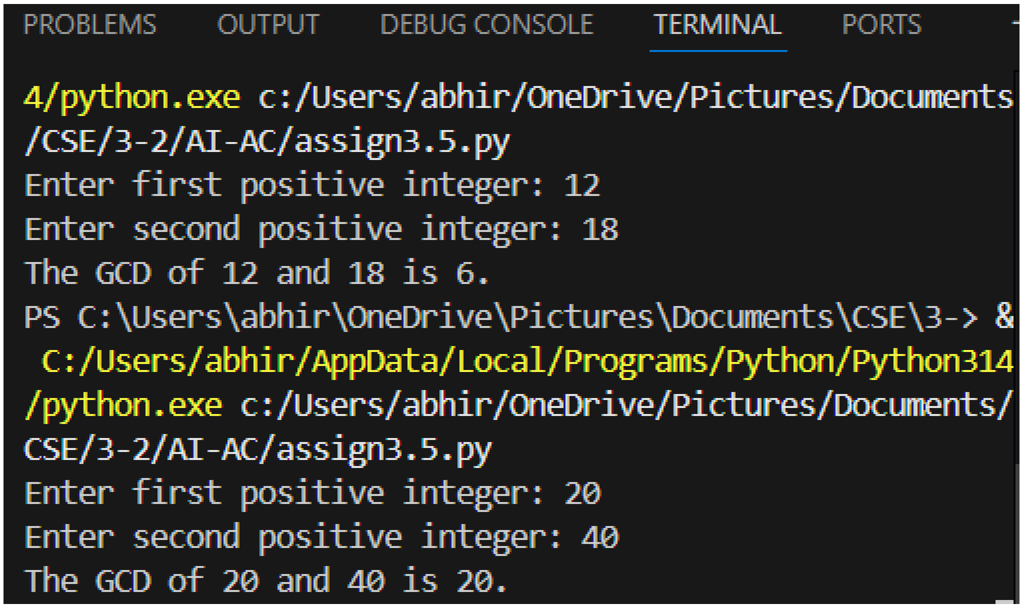
Code:

```
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.")
```



```
'''
a=12
b=18
display 6 as output
'''
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.")
```

Output :



```
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.
```

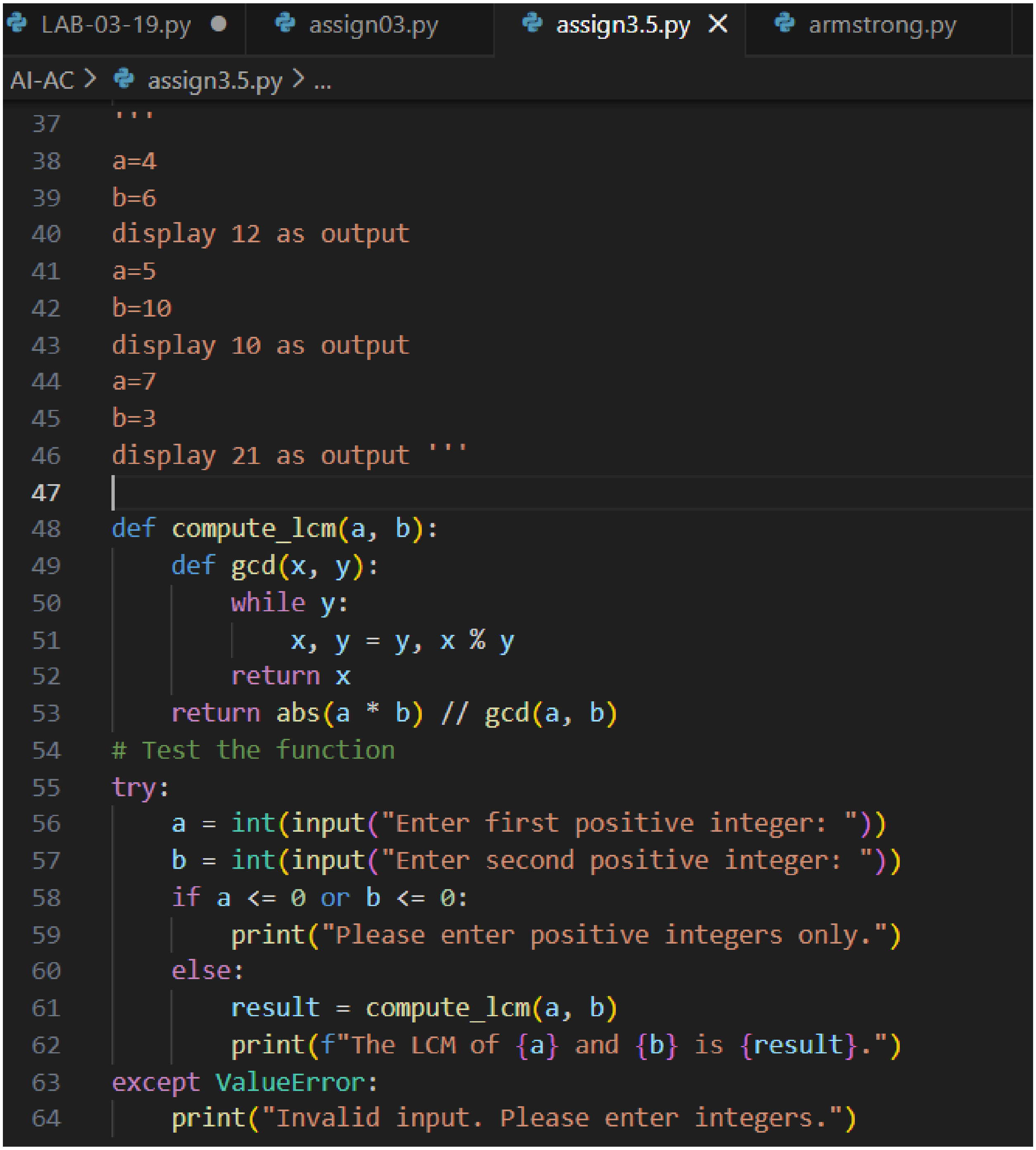
3) **Few-Shot Prompting (LCM Calculation) .Write a few-shot prompt with multiple examples to generate a Python function that computes the Least Common Multiple (LCM).**

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:

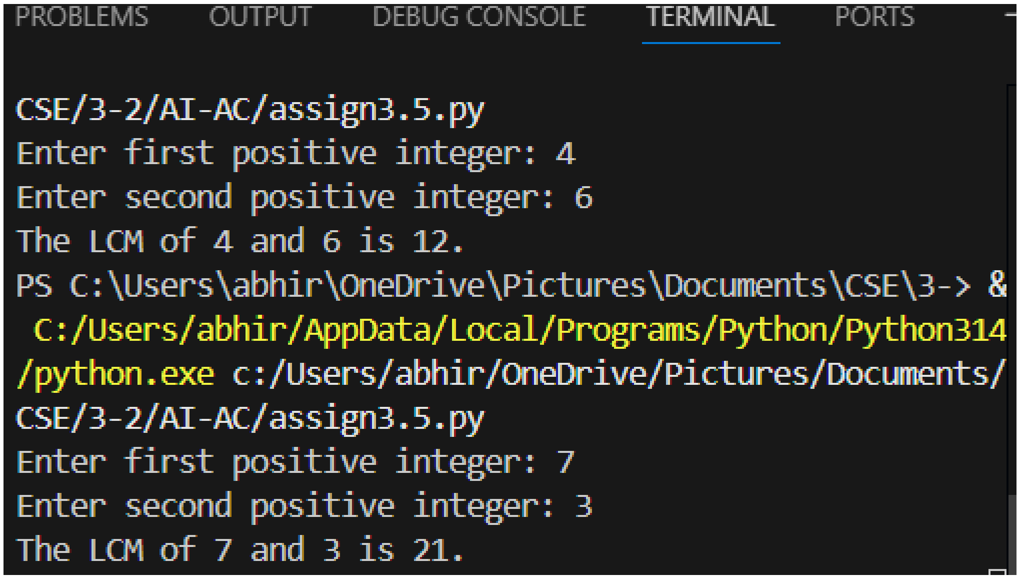
```
def compute_lcm(a, b):
    def gcd(x, y):
        while y:
            x, y = y, x % y
        return x
    return abs(a * b) // gcd(a, b)
# Test the function
try:
    a = int(input("Enter first positive integer: "))
    b = int(input("Enter second positive integer: "))
    if a <= 0 or b <= 0:
        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.")
```



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

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.")
```

Output:



```
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.**

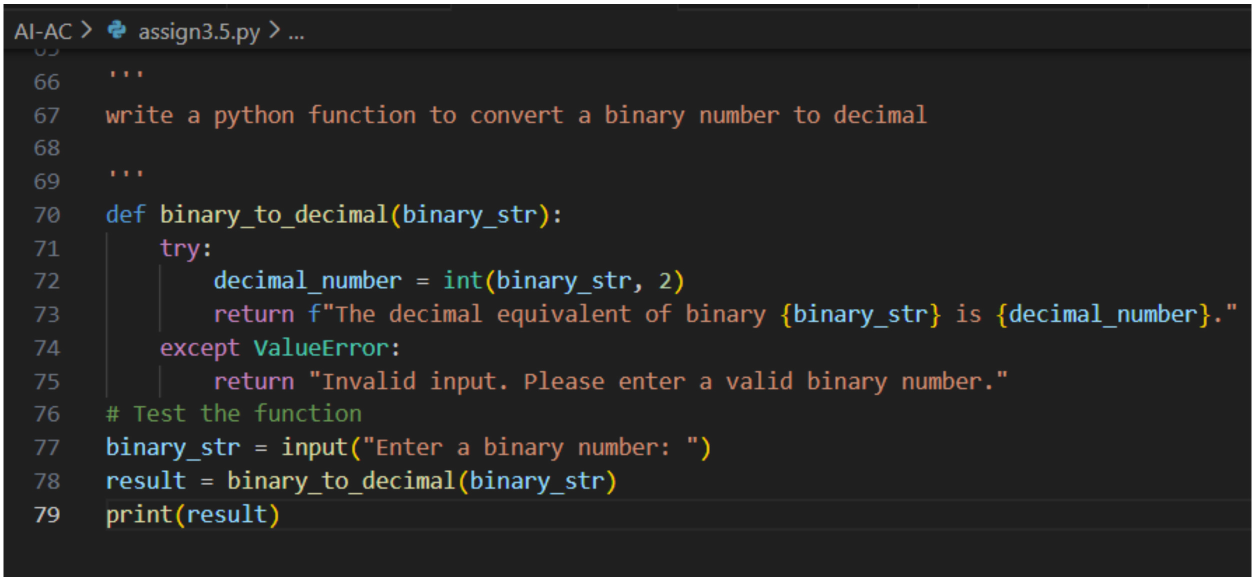
Prompt:

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)
```

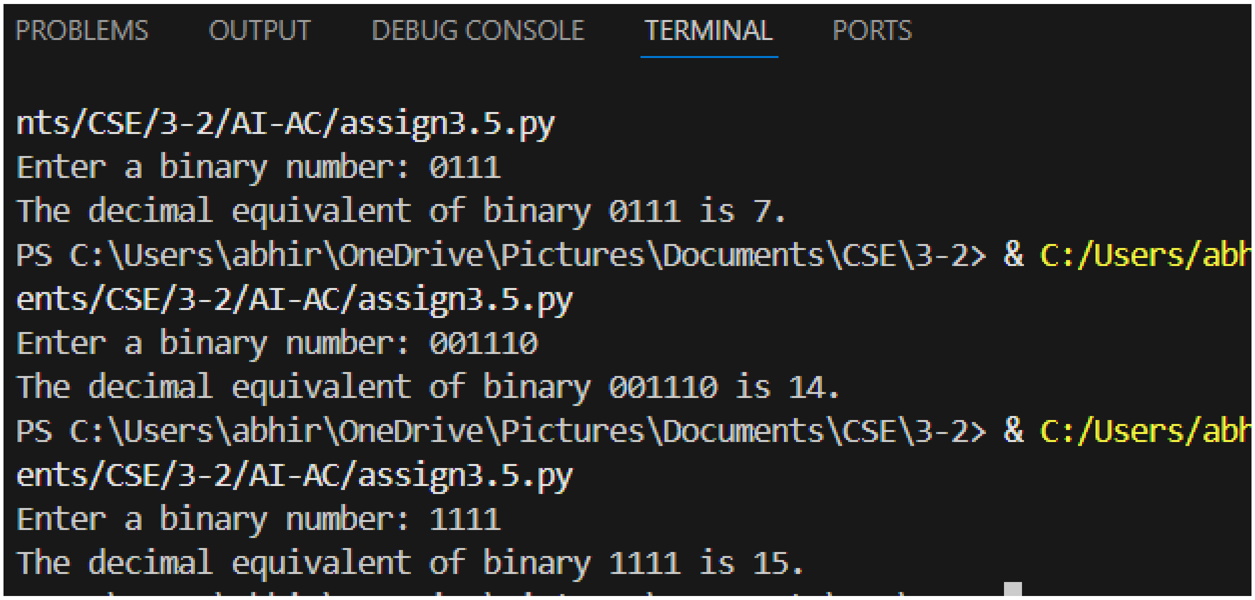
print(result)



```
'''
write a python function to convert a binary number to decimal

'''
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)
```

Output :



```
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/abh
ents/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/abh
ents/CSE/3-2/AI-AC/assign3.5.py
Enter a binary number: 1111
The decimal equivalent of binary 1111 is 15.
```

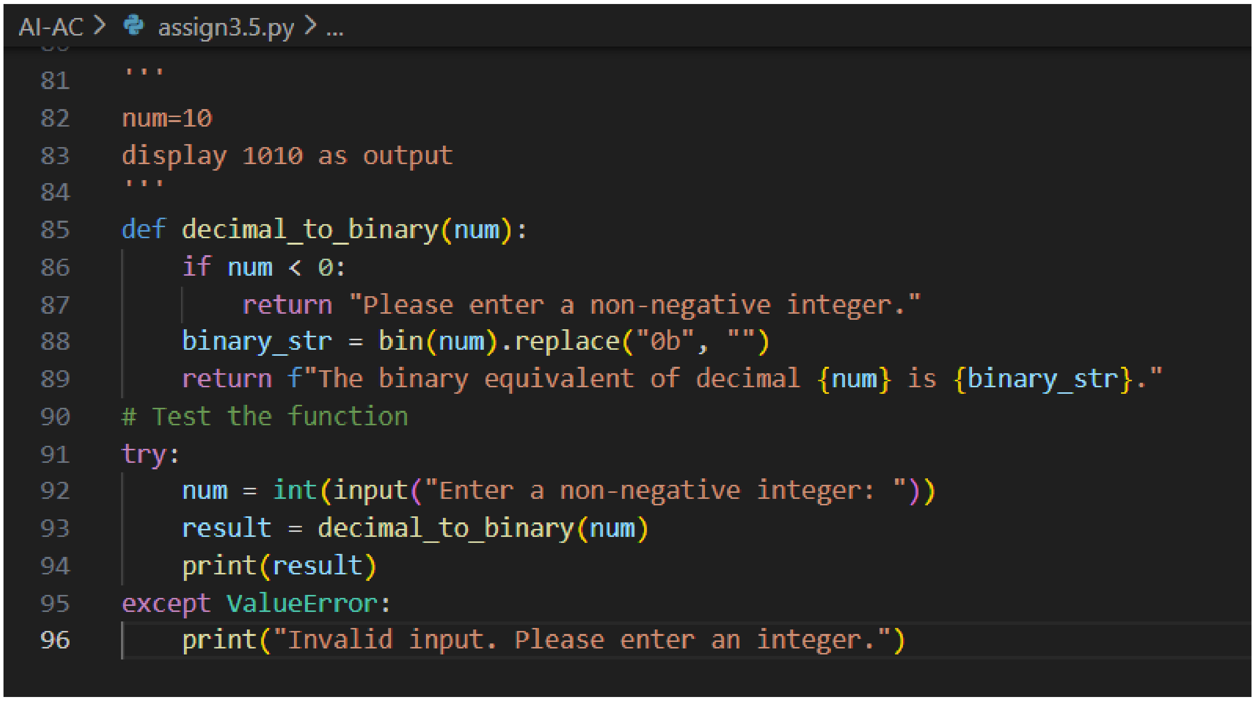
5) One-Shot Prompting (Decimal to Binary Conversion) . Write a one-shot prompt with an example to generate a Python function that converts a decimal number to binary.

Prompt :

num=10
display 1010 as output

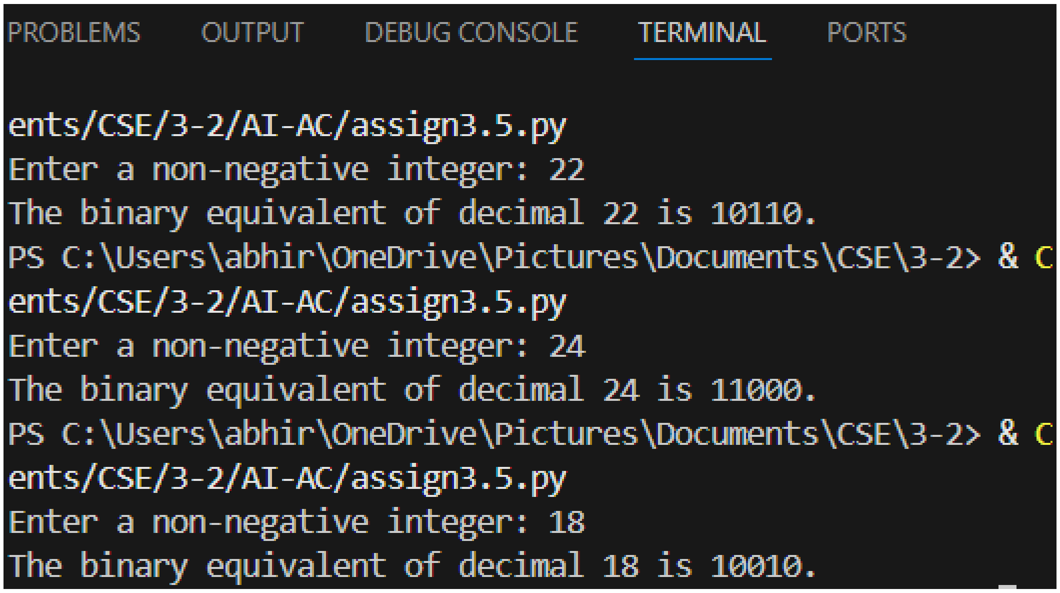
Code :

```
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.")
```



```
'''
num=10
display 1010 as output
'''
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.")
```

Output :



```
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.
```

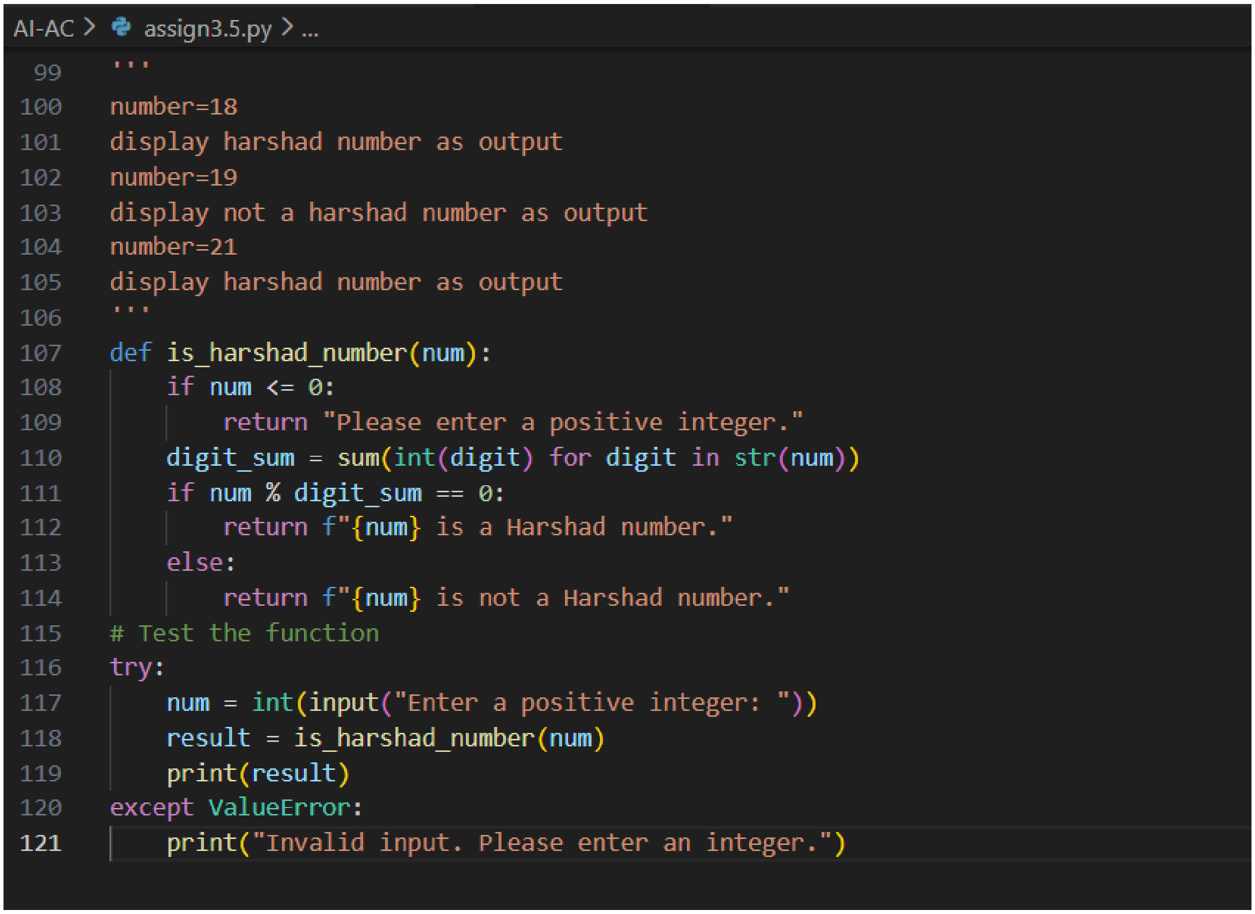
6) **Few-Shot Prompting (Harshad Number Check) . Write a few-shot prompt to generate a Python function that checkswhether a number is a Harshad (Niven) number.**

Prompt :

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

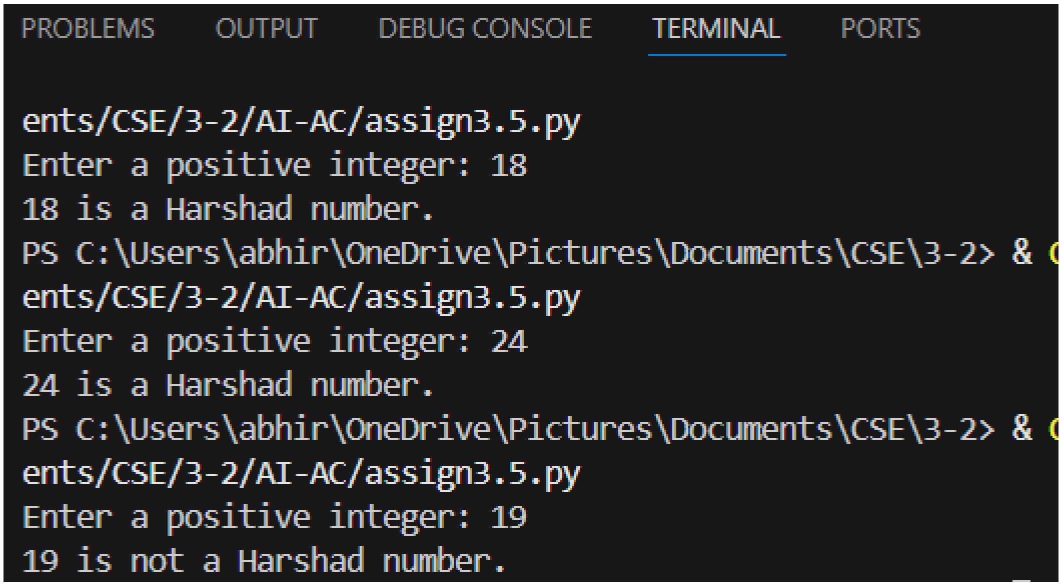
Code :

```
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.")
```



```python
'''
number=18
display harshad number as output
number=19
display not a harshad number as output
number=21
display harshad number as output
'''
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.")
```

Output :



```
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> &
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> &
ents/CSE/3-2/AI-AC/assign3.5.py
Enter a positive integer: 19
19 is not a Harshad number.
```