

## AIAC ASSIGNMENT – 3.5

G.Srija Reddy

2303A51671

BATCH-23

Prompt 1:

#write a zero shot to generate a python function that checks whether a year is leap or not

```
def is_leap_year(year):
```

```
    if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
```

```
        return True
```

```
    else:
```

```
        return False
```

# Example usage:

```
year = int(input("Enter a year: "))
```

```
print(f"{year} is a leap year: {is_leap_year(year)}")
```

output:

Enter a year: 1900

1900 is a leap year: False

Enter a year: 2000

2000 is a leap year: True

Enter a year: 2024

2024 is a leap year: True

Prompt 2:

#write a python code to find gcd of two numbers

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

```
    while b:
```

```
        a, b = b, a % b
```

```
    return a
def gcd_subtraction(a, b):
    while a != b:
        if a > b:
            a -= b
        else:
            b -= a
    return a
print(gcd(12,18))
```

```
print(gcd(48,18))
```

output:

6

6

Prompt 3:

#write a python code to generate a function that computes the least commom multiple (LCM) of two numbers

'''

input: 4,6-> output: 12

input: 5,10-> output: 10

input: 7,3-> output: 21

'''

```
def lcm(a, b):
    def gcd(x, y):
        while(y):
            x, y = y, x % y
```

```
    return x
```

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

```
# Example usage:
```

```
num1 = int(input("Enter first number: "))
```

```
num2 = int(input("Enter second number: "))
```

```
print(f"The LCM of {num1} and {num2} is: {lcm(num1, num2)}")
```

```
output:
```

```
Enter first number: 4
```

```
Enter second number: 6
```

```
The LCM of 4 and 6 is: 12
```

```
Enter first number: 5
```

```
Enter second number: 10
```

```
The LCM of 5 and 10 is: 10
```

```
Enter first number: 7
```

```
Enter second number: 3
```

```
The LCM of 7 and 3 is: 21
```

```
Prompt 4:
```

```
#Write a Python function that converts a binary number to decimal.
```

```
def binary_to_decimal(binary_str):
```

```
    decimal_value = 0
```

```
    binary_str = binary_str[::-1] # Reverse the string to process from least significant bit
```

```
    for index, digit in enumerate(binary_str):
```

```
        if digit == '1':
```

```
            decimal_value += 2 ** index
```

```
    return decimal_value
```

```
# Example usage:
```

```
binary_input = input("Enter a binary number: ")  
print(f"The decimal value is: {binary_to_decimal(binary_input)}")
```

output:

Enter a binary number: 1010

The decimal value is: 10

Prompt 5:

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

'''input:10->1010'''

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

# Example usage:

```
number = int(input("Enter a decimal number: "))  
print(f"The binary representation of {number} is: {decimal_to_binary(number)}")
```

output:

Enter a decimal number: 10

The binary representation of 10 is: 1010

