**Leap year or not**

def is\_leap\_year(year):

return (year%4==0) and (year % 100!=0 or year%400==0)

year = 2024

if is\_leap\_year(year):

print (f"{year} is a leap year.")

else:

print (f"{year} in not a leap year.")

**output:**2024 is a leap year.

**Multiplication table**

def print\_multiplication\_table (num, a) :

for i in range (1,a+1):

print (f"{num } \* {i} = {num \*i}")

print\_multiplication\_table (5,11)

**output:**

5 \* 1 = 5

5 \* 2 = 10

5 \* 3 = 15

5 \* 4 = 20

5 \* 5 = 25

5 \* 6 = 30

5 \* 7 = 35

5 \* 8 = 40

5 \* 9 = 45

5 \* 10 = 50

5 \* 11 = 55

**PRIME IN A RANGE:**

def is\_prime (n):

if n<=1:

return False

for i in range (2, int (n\*\*0.5)+1):

if n%i==0:

return False

return True

start = 10

end = 50

primes = [num for num in range (start, end +1) if is\_prime (num)]

print (f"Prime, numbers between {start} and {end }: { primes}")

**OUTPUT**: Prime, numbers between 10 and 50: [11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49]

**GCD AND LCM:**

import math

def compute\_gcd\_lcm (a,b):

gcd = math. gcd (a, b)

lcm = math. lcm (a,b)

return gcd, lcm

a=12

b=15

gcd,lcm=compute\_gcd\_lcm(a,b)

print (f"The GCD of {a} and {b} is {gcd}")

print (f"The LCM of {a} and {b} is {lcm}")

**OUTPUT:**

The GCD of 12 and 15 is 3

The LCM of 12 and 15 is 60

**FACTORIAL:**

def factorial\_iterative(n):

result = 1

for i in range(2, n + 1):

result \*= i

return result

n = 5

print(f"Factorial of {n} is {factorial\_iterative(n)}")

**PRIME NO OR NOT:**

def is\_prime(n):

if n <= 1:

return False

for i in range(2, n):

if n % i == 0:

return False

return True

n = 29

print(f"Is {n} a prime number? {is\_prime(n)}")

**OUTPUT:** Is 29 a prime number? True

**PALINDROME OR NOT:**

def is\_palindrome(s):

return s == s[::-1]

word = "racecar"

print(f"Is '{word}' a palindrome? {is\_palindrome(word)}")

**OUTPUT:**

Is 'racecar' a palindrome? True

**FIBONACCI:**

def fibonacci(n):

a, b = 0, 1

for \_ in range(n):

print(a, end=' ')

a, b = b, a + b

print()

fibonacci(10)

**OUTPUT:** 0 1 1 2 3 5 8 13 21 34

**SUM OF DIGITS:**

def sum\_of\_digits(n):

return sum(int(digit) for digit in str(n))

number = 1234

print(f"The sum of the digits of {number} is {sum\_of\_digits(number)}")

number = 56789

print(f"The sum of the digits of {number} is {sum\_of\_digits(number)}")

**OUTPUT:** The sum of the digits of 1234 is 10

The sum of the digits of 56789 is 35

**TECH NO:**

def count\_factors(n):

count = 0

for i in range(1, n + 1):

if n % i == 0:

count += 1

return count

def is\_tech\_number(n):

return count\_factors(n) % 2 != 0

num = 3025

if is\_tech\_number(num):

print(num, "is a tech number")

else:

print(num, "is not a tech number")

**OUTPUT:** 3025 is a tech number