**Day 01 python programs**

**#01**

**Progrm for palindrome or not**

Code:

def is\_palindrome(s):

return s == s[::-1]

# Test the function

word = "radar"

if is\_palindrome(word):

print(f"{word} is a palindrome")

else:

print(f"{word} is not a palindrome")

sample

input: Malayalam

output: Malayalam is a palindrome

**#02**

**Programe to find the given number is prime or not**

Code:

def is\_prime(num):

if num < 2:

return False

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

if num % i == 0:

return False

return True

# Test the function

number = 17

if is\_prime(number):

print(f"{number} is a prime number.")

else:

print(f"{number} is not a prime number.")

sample

input:7

output: 7 is not a prime number

**#03**

**Programe to the factorial of a number**

Code:

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n-1)

num = 5

print("Factorial of", num, "is", factorial(num))

sample

input:7

output: Factorial of 7 is 5040

**#04**

**Programe to print the fibbonacci series**

Code:

def fibonacci(n):

a, b = 0, 1

for \_ in range(n):

print(a, end=' ')

a, b = b, a + b

# Number of terms in the Fibonacci series

terms = 10

# Print the Fibonacci series

fibonacci(terms)

sample

input:4

output: 0 , 1, 1, 2, 3

**#05**

**Programe for sum of digits in a given number**

Code:

def sum\_of\_digits(number):

sum = 0

while number > 0:

digit = number % 10

sum += digit

number //= 10

return sum

number = 12345

print("The sum of digits in", number, "is:", sum\_of\_digits(number))

sample

input:786

output:21

**#06**

**Programe to print multiplication table**

Code:

def multiplication\_table(num):

for i in range(1, 11):

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

# Generate multiplication table for a specific number

number = 5

multiplication\_table(number)

sample

input: 3

output:

3 x 1 = 3

3 x 2= 6

3 x 3 = 9

3 x 4 = 12

3 x 5 = 15

3 x 6 = 18

3 x 7 = 21

3 x 8 = 24

3 x 9 = 27

3 x 10 = 30

**#07**

**Program for LCM and GCD**

Code:

def compute\_lcm(x, y):

if x > y:

greater = x

else:

greater = y

while True:

if greater % x == 0 and greater % y == 0:

lcm = greater

break

greater += 1

return lcm

def compute\_gcd(x, y):

while y:

x, y = y, x % y

return x

num1 = 54

num2 = 24

print("The LCM of", num1, "and", num2, "is", compute\_lcm(num1, num2))

print("The GCD of", num1, "and", num2, "is", compute\_gcd(num1, num2))

1. input: 1 ,3
2. ouptput:
3. lcm:3
4. gcd:1

**#08**

**Progrme to detect a tech number or not**

Code:

def is\_tech\_number(num):

num\_str = str(num)

for i in range(len(num\_str) - 2):

if int(num\_str[i]) + int(num\_str[i+1]) == int(num\_str[i+2]):

continue

else:

return False

return True

# Test the function

number = 124

if is\_tech\_number(number):

print(f"{number} is a Tech Number.")

else:

print(f"{number} is not a Tech Number.")

input:81

output: 81 is a tech number

**#09**

**Program for prime in given range**

Code:

def is\_prime(num):

if num < 2:

return False

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

if num % i == 0:

return False

return True

def generate\_primes\_in\_range(start, end):

prime\_numbers = []

for num in range(start, end + 1):

if is\_prime(num):

prime\_numbers.append(num)

return prime\_numbers

start\_range = 10

end\_range = 50

prime\_numbers\_in\_range = generate\_primes\_in\_range(start\_range, end\_range)

print("Prime numbers between", start\_range, "and", end\_range, "are:", prime\_numbers\_in\_range)

input:

lower:1

upper:10

output:

2 3 5 7

**#10**

Program **to detect a leap year or not**

Code:

def is\_leap\_year(year):

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

return True

else:

return False

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

if is\_leap\_year(year):

print(year, "is a leap year.")

else:

print(year, "is not a leap year.")

input:2004

output: 2004 is a leap year.