**Python program to find GCD of a number:**

def gcd(a, b):

while b:

a, b = b, a % b

return a

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

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

result = gcd(num1, num2)

print(f"The GCD of {num1} and {num2} is {result}")

**Python program to find factorial of a number using recursion:**

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n - 1)

number = int(input("Enter a non-negative integer: "))

result = factorial(number)

print(f"The factorial of {number} is {result}")

Here's how the recursive calculation proceeds when calculating factorial(4):

factorial(4) calls factorial(3)

factorial(3) calls factorial(2)

factorial(2) calls factorial(1)

factorial(1) calls factorial(0)

factorial(0) reaches the base case and returns 1

factorial(1) returns 1 \* 1 = 1

factorial(2) returns 2 \* 1 = 2

factorial(3) returns 3 \* 2 = 6

factorial(4) returns 4 \* 6 = 24

**Python program to check whether a year is leap year or not**

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

else:

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

**Design the flowchart and write code that calculates salary of an employee. Prompt the user to enter the basic salary, HRA TA and DA. Add these components to calculate the gross salary and also deduce 10% salary from the gross salary to be paid as tax.**

Flowchart Description:

Start

Prompt the user to enter basic\_salary, HRA (House Rent Allowance), TA (Travel Allowance), and DA (Dearness Allowance).

Calculate the gross\_salary by summing up basic\_salary, HRA, TA, and DA.

Calculate tax as 10% of gross\_salary.

Calculate net\_salary as gross\_salary - tax.

Display gross\_salary, tax, and net\_salary.

End

**Python Code:**

# Step 2: Prompt the user to enter the components

basic\_salary = float(input("Enter the basic salary: "))

HRA = float(input("Enter the House Rent Allowance (HRA): "))

TA = float(input("Enter the Travel Allowance (TA): "))

DA = float(input("Enter the Dearness Allowance (DA): "))

# Step 3: Calculate the gross salary

gross\_salary = basic\_salary + HRA + TA + DA

# Step 4: Calculate tax

tax = 0.1 \* gross\_salary # 10% of gross salary

# Step 5: Calculate net salary

net\_salary = gross\_salary - tax

# Step 6: Display the results

print(f"Gross Salary: {gross\_salary}")

print(f"Tax: {tax}")

print(f"Net Salary: {net\_salary}")

**Exponent of a number using recursion:**

def power(base, exponent):

if exponent == 0:

return 1

else:

return base \* power(base, exponent - 1)

base = float(input("Enter the base number: "))

exponent = int(input("Enter the exponent: "))

result = power(base, exponent)

print(f"{base} raised to the power of {exponent} is {result}")

In this code, the power function is defined recursively. It has a base case (when exponent is 0) that returns 1, and a recursive case (when exponent is greater than 0) that calculates base times power(base, exponent - 1).

For example, if you input base = 2 and exponent = 3, the recursive calculation proceeds as follows:

power(2, 3) calls power(2, 2)

power(2, 2) calls power(2, 1)

power(2, 1) calls power(2, 0)

power(2, 0) reaches the base case and returns 1

power(2, 1) returns 2 \* 1 = 2

power(2, 2) returns 2 \* 2 = 4

power(2, 3) returns 2 \* 4 = 8

The final result, in this case, is 8, which is 2 raised to the power of 3.