• Write a program to find power of any number using recursion.

• Write a program to print all natural numbers between 1 to n using recursion.

• Write a program to print all even or odd numbers in given range using recursion.

• Write a program to find sum of all natural numbers between 1 to n using recursion.

• Write a program to find reverse of any number using recursion.

• Write a program to find sum of digits of a given number using recursion.

• Write a program to find factorial of any number using recursion

• Write a program to generate nth Fibonacci term using recursion.

• Write a program to find GCD (HCF) of two numbers using recursion.

**Answers -**

1. Program to find power of any number using recursion:

python

def power(base, exponent):

if exponent == 0:

return 1

else:

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

# Example usage

base = 2

exponent = 3

print("Result:", power(base, exponent)) # Output: 8

2. Program to print all natural numbers between 1 to n using recursion:

python

def print\_natural\_numbers(n):

if n > 0:

print\_natural\_numbers(n - 1)

print(n)

# Example usage

n = 5

print\_natural\_numbers(n) # Output: 1 2 3 4 5

3. Program to print all even or odd numbers in a given range using recursion:

python

def print\_even\_or\_odd(start, end, mode):

if start <= end:

if mode == 'even' and start % 2 == 0:

print(start)

elif mode == 'odd' and start % 2 != 0:

print(start)

print\_even\_or\_odd(start + 1, end, mode)

# Example usage

start = 1

end = 10

mode = 'even'

print\_even\_or\_odd(start, end, mode) # Output: 2 4 6 8 10

4. Program to find sum of all natural numbers between 1 to n using recursion:

python

def sum\_natural\_numbers(n):

if n == 0:

return 0

else:

return n + sum\_natural\_numbers(n - 1)

# Example usage

n = 5

print("Sum:", sum\_natural\_numbers(n)) # Output: 15

5. Program to find reverse of any number using recursion:

python

def reverse\_number(n):

if n < 10:

return n

else:

return int(str(n % 10) + str(reverse\_number(n // 10)))

# Example usage

number = 12345

print("Reversed number:", reverse\_number(number)) # Output: 54321

6. Program to find sum of digits of a given number using recursion:

python

def sum\_of\_digits(n):

if n < 10:

return n

else:

return n % 10 + sum\_of\_digits(n // 10)

# Example usage

number = 12345

print("Sum of digits:", sum\_of\_digits(number)) # Output: 15

7. Program to find factorial of any number using recursion:

python

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n - 1)

# Example usage

number = 5

print("Factorial:", factorial(number)) # Output: 120

8. Program to generate nth Fibonacci term using recursion:

python

def fibonacci(n):

if n <= 1:

return n

else:

return fibonacci(n - 1) + fibonacci(n - 2)

# Example usage

n = 6

print("Fibonacci term:", fibonacci(n)) # Output: 8

9. Program to find GCD (HCF) of two numbers using recursion:

python

def gcd(a, b):

if b == 0:

return a

else:

return gcd(b, a % b)

# Example usage

num1 = 48

num2 = 18

print("GCD:", gcd(num1, num2)) # Output: 6