**1.Write a program to Print the Fibonacci Series using recursion.**

**Program:**

def fibonacci(n):

if n <= 1:

return n

else:

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

nterms = 10

if nterms <= 0:

print("Please enter a positive integer")

else:

print("Fibonacci sequence:")

for i in range(nterms):

print(fibonacci(i))

**2.Write a program to check whether the given no is Armstrong or not using a recursive function**.

**Program**:

def order(n):

return len(str(n))

def is\_armstrong(n, order):

if n == 0:

return 0

else:

return (n % 10) \*\* order + is\_armstrong(n // 10, order)

def check\_armstrong(n):

order\_val = order(n)

temp = is\_armstrong(n, order\_val)

if temp == n:

return True

else:

return False

num = int(input("Enter a number: "))

if check\_armstrong(num):

print(num, "is an Armstrong number.")

else:

print(num, "is not an Armstrong number.")

**3.** **Write a program to find the GCD of two numbers using recursive factorization**

**Program:**

def gcd(a, b):

if b == 0:

return a

else:

return gcd(b, a % b)

num1 = 48

num2 = 18

result = gcd(num1, num2)

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

**4. Write a program to get the largest element of an array.**

**Program:**

def get\_largest\_element(arr):

return max(arr)

array = [10, 5, 20, 8]

largest\_element = get\_largest\_element(array)

print("The largest element in the array is:", largest\_element)

**5. Write a program to find the Factorial of a number using recursion.**

**Program:**

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n-1)

num = 5

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

**6. Write a program to copy one string to another using recursion**

**Program:**

def copy\_string(source, destination, index=0):

if index == len(source):

return destination

destination += source[index]

return copy\_string(source, destination, index + 1)

source\_str = "Hello, World!"

destination\_str = ""

result = copy\_string(source\_str, destination\_str)

print(result)

**7. Write a program to print the reverse of a string using recursion.**

**Program:**

def reverse\_string(s):

if len(s) == 0:

return s

else:

return reverse\_string(s[1:]) + s[0]

input\_string = "Hello, World!"

reversed\_string = reverse\_string(input\_string)

print("Original String:", input\_string)

print("Reversed String:", reversed\_string)

**8. Write a program to generate all the prime numbers using recursion.**

**Program:**

def is\_prime(n, i=2):

if n <= 2:

return True if n == 2 else False

if n % i == 0:

return False

if i \* i > n:

return True

return is\_prime(n, i + 1)

def generate\_primes(n):

if n > 1:

generate\_primes(n - 1)

if is\_prime(n):

print(n)

n = int(input("Enter a number: "))

print(f"Prime numbers up to {n}:")

generate\_primes(n)

**9. Write a program to check whether a number is a prime number or not using recursion.**

**Program:**

def is\_prime(num, div=2):

if num <= 2:

return num == 2

if num % div == 0:

return False

if div \* div > num:

return True

return is\_prime(num, div + 1)

num = int(input("Enter a number: "))

if is\_prime(num):

print(num, "is a prime number")

else:

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

**10. Write a program to check whether a given String is Palindrome or not using recursion.**

**Program:**

def is\_palindrome(s):

s = s.lower().replace(" ", "")

if len(s) <= 1:

return True

if s[0] != s[-1]:

return False

return is\_palindrome(s[1:-1])

input\_string = "A man a plan a canal Panama"

if is\_palindrome(input\_string):

print(f"{input\_string} is a palindrome.")

else:

print(f"{input\_string} is not a palindrome.")