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

def fibonacci(n):

if n <= 1:

return n

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

def print\_fibonacci\_series(n):

for i in range(n):

print(fibonacci(i), end=' ')

print() # for newline

# Example usage

n\_terms = 10 # You can change this value to generate more terms

print\_fibonacci\_series(n\_terms)

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

def is\_armstrong(n, power=None):

if power is None:

power = len(str(n))

if n == 0:

return 0

return (n % 10) \*\* power + is\_armstrong(n // 10, power) if power else n

# Example usage

number = 153 # You can change this value to test other numbers

if number == is\_armstrong(number):

print(f"{number} is an Armstrong number.")

else:

print(f"{number} is not an Armstrong number.")

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

def gcd(a, b):

if b == 0:

return a

return gcd(b, a % b)

# Example usage

num1 = 48

num2 = 18

result = gcd(num1, num2)

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

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

def find\_largest(arr):

if not arr:

return float('-inf')

return max(arr[0], find\_largest(arr[1:]))

# Example usage

array = [3, 5, 7, 2, 8, 6]

largest\_element = find\_largest(array)

print(f"The largest element in the array is {largest\_element}.")

1. **Write a program to find the Factorial of a number using recursion.6. Write a program for to copy one string to another using recursion**

def factorial(n):

return 1 if n in (0, 1) else n \* factorial(n - 1)

# Example usage

number = 5 # You can change this value to test other numbers

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

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

def reverse(s):

if len(s) == 0:

return s

else:

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

s = str(input())

print("The original string is : ", end="")

print(s)

print("The reversed string is : ", end="")

print(reverse(s))

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

def prime(x, y):

prime\_list = []

for i in range(x, y):

if i == 0 or i == 1:

continue

else:

for j in range(2, int(i/2)+1):

if i % j == 0:

break

else:

prime\_list.append(i)

return prime\_list

start=int(input("enter starting no"))

end= int(input("enter ending no"))

lst = prime(start,end)

if len(lst) == 0:

print("There are no prime numbers in this range")

else:

print("The prime numbers in this range are: ", lst)

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

def 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 Prime(n, i + 1)

n=int(input())

if (Prime(n)):

print("Yes")

else:

    print("No")

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

def palindrome(s):

if len(s) < 1:

return True

else:

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

return palindrome(s[1:-1])

else:

return False

a=str(input("Enter string:"))

if(palindrome(a)==True):

print("String is a palindrome")

else:

print("String isn't a palindrome")