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

def fibonacci(n):

if n <= 1:

return n

else:

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

def print\_fibonacci\_series(length):

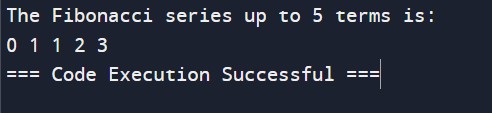
for i in range(length):

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

n = 5

print(f"The Fibonacci series up to {n} terms is:")

print\_fibonacci\_series(n)



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

def count\_digits(num):

if num == 0:

return 0

return 1 + count\_digits(num // 10)

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

if n is None:

n = count\_digits(num)

if num == 0:

return 0

else:

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

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

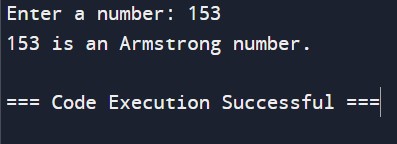
sum\_of\_powers = is\_armstrong(number)

if sum\_of\_powers == 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\_recursive(a, b):

if b == 0:

return a

else:

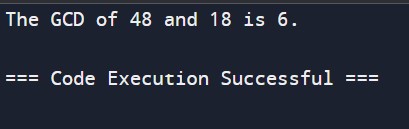
return gcd\_recursive(b, a % b)

num1 = 48

num2 = 18

result = gcd\_recursive(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\_element(arr):

if len(arr) == 0:

return None

max\_element = arr[0]

for element in arr[1:]:

if element > max\_element:

max\_element = element

return max\_element

array = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

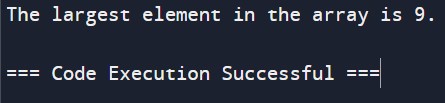
largest\_element = find\_largest\_element(array)

if largest\_element is not None:

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

else:

print("The array is empty.")



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

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n-1)

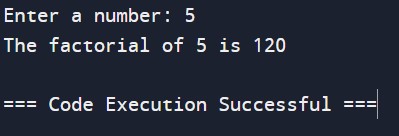
num = int(input("Enter a number: "))

if num < 0:

print("Factorial is not defined for negative numbers.")

else:

print(f"The factorial of {num} is {factorial(num)}")



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

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

if index == len(source):

return destination

destination += source[index]

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

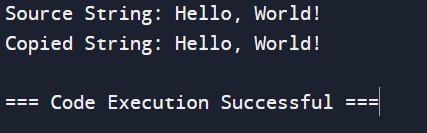
source\_str = "Hello, World!"

destination\_str = ""

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

print("Source String:", source\_str)

print("Copied String:", result)



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

def reverse\_string(s):

if len(s) == 0:

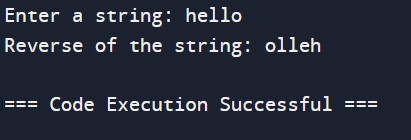
return s

else:

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

input\_string = input("Enter a string: ")

print("Reverse of the string:", reverse\_string(input\_string))



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

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

if n <= 1:

return False

elif divisor > n // 2:

return True

elif n % divisor == 0:

return False

else:

return is\_prime(n, divisor + 1)

def generate\_primes(limit, current=2, primes=[]):

if current > limit:

return primes

elif is\_prime(current):

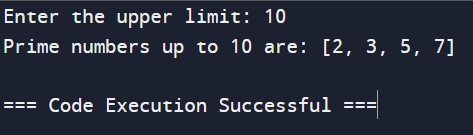
primes.append(current)

return generate\_primes(limit, current + 1, primes)

limit = int(input("Enter the upper limit: "))

prime\_numbers = generate\_primes(limit)

print("Prime numbers up to", limit, "are:", prime\_numbers)



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

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

if n <= 1:

return False

elif n == divisor:

return True

elif n % divisor == 0:

return False

else:

return is\_prime(n, divisor + 1)

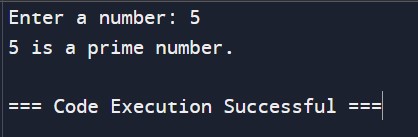
num = int(input("Enter a number: "))

if is\_prime(num):

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

else:

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



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

def is\_palindrome(s):

s = s.lower()

if len(s) <= 1:

return True

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

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

else:

return False

input\_string = input("Enter a string: ")

if is\_palindrome(input\_string):

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

else:

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

