**1. Fibonacci Series**

# Python

def fibonacci(n):

print("Incorrect input")

return

elif n == 0:

return 0

elif n == 1:

return 1

else:

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

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

**2. Armstrong Number**

# Python

def is\_armstrong(number): original\_number = number

sum = 0

num\_of\_digits = len(str(number))

while number > 0:

digit = number % 10

sum += digit \*\* num\_of\_digits

number //= 10

return sum == original\_number

num = 153

if is\_armstrong(num):

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

else:

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

**3. Greatest Common Divisor (GCD)**

# Python

def gcd(a, b):

if b == 0:

return a

else:

return gcd(b, a % b)

x = 30 y=45

gcd\_value = gcd(x, y)

print(f"GCD of {x} and {y} is {gcd\_value}")

**4. Largest Element in Array**

def largest\_element(arr, i):

if i == len(arr) - 1:

return arr[i]

else:

largest = largest\_element(arr, i + 1)

return

largest if largest > arr[i] else arr[i]

arr = [10, 25, 12, 3, 70]

largest = largest\_element(arr, 0)

print(f"Largest element in the array is {largest}")

**5. Factorial**

# Python

def factorial(n):

if n == 0: return 1 else:

return n \* factorial(n-1)

num = 5

fact = factorial(num)

print(f"Factorial of {num} is {fact}")

**6. String Copy**

# Python

def copy\_string(source, dest, i):

return

else:

dest[i] = source[i]

copy\_string(source, dest, i + 1)

source\_str = "Hello"

dest\_str = [None] \* len(source\_str) + ["\0"]

null terminator

copy\_string(source\_str, dest\_str, 0)

print(f"Copied string: {''.join(dest\_str[:-1

**7. String Reverse**

# Python

def reverse\_string(string, i):

if i == len(string) // 2:

return else:

temp = string[i]

string[i] = string[len(string) - i - 1] string[len(string) - i - 1] = temp reverse\_string(string, i + 1)

text = "World" reverse\_string(text, 0) print(f"Reversed string: {text}")

**8.Prime Numbers**

def sieve\_of\_eratosthenes(n):

primes = [True] \* (n + 1)

primes[0] = primes[1] = False # 0 and 1 are not prime for i in range(2, int(n\*\*0.5) + 1):

if primes[i]: for j in range(i \* i, n + 1, i):

primes[j] = False

return [i for i, is\_prime in enumerate(primes) if is\_prime]

limit = 20

primes = sieve\_of\_eratosthenes(limit) print(f"Prime numbers up to {limit}: {primes}")

**9. Check Prime Number**

# Python

def is\_prime(n):

if n <= 1:

return False

elif n <= 3:

return True

elif n % 2 == 0 or n % 3 == 0:

return False

i = 5

while i \* i <= n:

if n % i == 0 or n % (i + 2) == 0:

return False

i += 6

return True

num = 11

if is\_prime(num):

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

else:

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

**10. Palindrome Check**

# Python

def is\_palindrome(string, start, end):

if start >= end:

return True

elif string[start] != string[end]:

return False

else:

return is\_palindrome(string, start + 1, end - 1)

text = "racecar"

if is\_palindrome(text, 0, len(text) - 1):

print(f"'{text}' is a palindrome")

else:

print(f"'{text}' is not a palindrome")