# Day 1 programs

1.Fibonacci series using recursion:

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.Armstrong number or not using recursion:

def order(n):

count = 0

while n != 0:

count += 1

n //= 10

return count

def is\_armstrong(n, order\_val):

if n == 0:

return 0

else:

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

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

order\_val = order(num)

if num == is\_armstrong(num, order\_val):

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

else:

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

3.GCD of two numbers using recursion:

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}")

4.Largest element of an array:

def find\_largest\_element(arr):

max\_element = arr[0]

for i in range(1, len(arr)):

if arr[i] > max\_element:

max\_element = arr[i]

return max\_element

array = [10, 30, 20, 50, 40]

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

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

5.Factorial of a number using recursion:

def factorial(n):

if n == 0:

return 1

else:

return n \* factorial(n-1)

num = 5

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

6.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)

7.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 = "Hello, World!"

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

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

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

8.generate all the prime numbers using recursion:

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)

generate\_primes(20)

9.prime or not using recursion:

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

if num <= 2:

return num == 2

if num % i == 0:

return False

if i \* i > num:

return True

return is\_prime(num, i + 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. String palindrome or not using recursion:

def is\_palindrome(s):

s = s.lower()

s = ''.join(e for e in s if e.isalnum())

if len(s) < 2:

return True

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

return False

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

string = "A man, a plan, a canal, Panama"

if is\_palindrome(string):

print(f"{string} is a palindrome.")

else:

print(f"{string} is not a palindrome.")