Day 1 lab programs

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)

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 the given no is Armstrong or not using recursive function.

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. 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}")

4. Write a program to get the 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. 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):

if n == 0:

return 1

else:

return n \* factorial(n-1)

number = 5

result = factorial(number)

print(f'The factorial of {number} is: {result}')

7. 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 = "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

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)

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

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

if num <= 1:

return False

if div \* div > num:

return True

if num % div == 0:

return False

return is\_prime(num, div + 1)

# Input number to check for primality

number = 17

if is\_prime(number):

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

else:

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

10. Write a program for to check whether a given String is 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])

# Test the function

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.")