Lab

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

Code:-

def fibonacci(n):

if n <= 0:

return "Input should be a positive integer"

elif n == 1:

return 0

elif n == 2:

return 1

else:

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

for i in range(1, 11):

print(fibonacci(i))

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

Code:-

def count\_digits(n):

if n == 0:

return 0

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

def check\_armstrong(n, count):

if n == 0:

return 0

digit = n % 10

return (digit \*\* count) + check\_armstrong(n // 10, count)

def is\_armstrong(n):

count = count\_digits(n)

return n == check\_armstrong(n, count)

num = 153

if is\_armstrong(num):

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

else:

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

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

Code:-

def gcd(a, b):

if b == 0:

return a

else:

return gcd(b, a % b)

num1 = 60

num2 = 48

print(f"The GCD of {num1} and {num2} is {gcd(num1, num2)}")

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

Code:-

def find\_largest(arr):

if len(arr) == 0:

return None

max\_element = arr[0]

for element in arr:

if element > max\_element:

max\_element = element

return max\_element

arr = [10, 20, 4, 45, 99]

print(f"The largest element in the array is {find\_largest(arr)}")

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

Code:-

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n - 1)

num = 5

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

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

Code:-

def copy\_string(src, dest = ''):

if src == '':

return dest

else:

return copy\_string(src[1:], dest + src[0])

src = "Hello, World!"

print(f"Original string: {src}")

print(f"Copied string: {copy\_string(src)}")

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]

str = "Hello, World!"

print(f"Original string: {str}")

print(f"Reversed string: {reverse\_string(str)}")

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

Code:-

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 print\_primes(n):

if n > 1:

print\_primes(n - 1)

if is\_prime(n):

print(n)

print\_primes(20)

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

Code:-

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)

# Test the function

num = 17

if is\_prime(num):

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

else:

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

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

Code:-

def is\_palindrome(s):

if len(s) <= 1:

return True

else:

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

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

else:

return False

str = "madam"

if is\_palindrome(str):

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

else:

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