EXPERIMENT 1-10

-Prisha D(192311018)

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

```
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)

def print_fibonacci_series(count):
    for i in range(1, count+1):
        print(fibonacci(i), end=" ")

# Example usage:
n_terms = 10  # Number of terms in the Fibonacci series to print
print_fibonacci_series(n_terms)
```

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

```
def is_armstrong(n, temp=None, sum_powers=0):
    if temp is None:
        temp = n
    if temp == 0:
        return sum_powers == n
    digit = temp % 10
    num_of_digits = len(str(n))
```

```
    return is_armstrong(n, temp // 10, sum_powers + digit ** num_of_digits)

# Example usage:
number = 153
if is_armstrong(number):
    print(f"{number} is an Armstrong number.")
else:
    print(f"{number} is not an Armstrong number.")
```

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

```
def gcd(a, b):
    if b == 0:
        return a
    else:
        return gcd(b, a % b)

# Example usage:
num1 = 48
num2 = 18
result = gcd(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(arr):
    if len(arr) == 1:
        return arr[0]
    else:
        max_of_rest = find_largest(arr[1:])
        return arr[0] if arr[0] > max_of_rest else max_of_rest

# Example usage:
```

```
array = [3, 1, 4, 1, 5, 9, 2, 6, 5]
largest_element = find_largest(array)
print(f"The largest element in the array is {largest_element}.")
```

5. 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)

# Example usage:
number = 5
print(f"The factorial of {number} is {factorial(number)}.")
```

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

```
def copy_string(src, dst="", index=0):
    if index == len(src):
        return dst
    else:
        return copy_string(src, dst + src[index], index + 1)

# Example usage:
original_string = "Hello, World!"
copied_string = copy_string(original_string)
print(f"Original string: {original_string}")
print(f"Copied string: {copied_string}")
```

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]

# Example usage:
original_string = "Hello, World!"
reversed_string = reverse_string(original_string)
print(f"Reversed string: {reversed_string}")
```

8. Write a program to 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, current=2, primes=[]):
    if current > n:
        return primes
    if is_prime(current):
        primes.append(current)
    return generate_primes(n, current + 1, primes)

# Example usage:
n = 20
primes_up_to_n = generate_primes(n)
print(f"Prime numbers up to {n}: {primes_up_to_n}")
```

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

# Example usage:
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):
    if len(s) <= 1:
        return True
    if s[0] != s[-1]:
        return False
    return is_palindrome(s[1:-1])

# Example usage:
string = "racecar"
if is_palindrome(string):
    print(f"{string} is a palindrome.")
else:
    print(f"{string} is not a palindrome.")
```