

**1. Write a program to find sum of following series using recursive functions.**

```
def factorial(n):
    if n == 1 or n == 0:
        return 1
    return n * factorial(n - 1)

def sum_of_series(n):
    if n == 1:
        return 1
    return factorial(n) + sum_of_series(n - 1)

n = int(input("Enter n:"))
print(f"{sum_of_series(n)}")
```

**2. Write a program to check if given number is Armstrong or not using recursive function.**

```
def armstrong_sum(n, digits):
    if n == 0:
        return 0
    return (n % 10) ** digits + armstrong_sum(n // 10, digits)

def is_armstrong(n):
    digits = len(str(n))
    return n == armstrong_sum(n, digits)

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

if is_armstrong(num):
    print("Armstrong number")
else:
    print("Not an Armstrong number")
```

**3. Write a program to reverse a given number using recursive function.**

```
def reverse(n):
    if n < 10:
        return n
    return int(str(n % 10) + str(reverse(n // 10)))

num = int(input("Enter a number: "))
print(f"{reverse(num)}")
```

**4. Write a program to find sum of n numbers using recursion.**

```
def sum_n(n):
    if n == 1:
        return 1
    return n + sum_n(n - 1)

num = int(input("Enter a number: "))
print(f"{sum_n(num)}")
```

**5. Write a program to find factorial using recursion.**

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    return n * factorial(n - 1)

num = int(input("Enter a number: "))
print(f"{factorial(num)}")
```

**6. Write a program to print Fibonacci series using recursion.**

```
def fibonacci(n):
    if n <= 1:
        return n
    return fibonacci(n - 1) + fibonacci(n - 2)

num = int(input("Enter the number of terms: "))
print("Fibonacci Series:", [fibonacci(i) for i in range(num)])
```

**7. Write a program to find sum of digits using recursion.**

```
def sum_of_digits(n):
    if n == 0:
        return 0
    return (n % 10) + sum_of_digits(n // 10)

num = int(input("Enter a number: "))
print(f"{sum_of_digits(num)}")
```

**8. Write a program to check whether a number is prime or not using recursion.**

```
def is_prime(n, i=2):
    if n <= 1:
        return False
    if i * i > n:
        return True
    if n % i == 0:
        return False
    return is_prime(n, i + 1)

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

if is_prime(num):
    print("Prime number")
else:
    print("Not a prime number")
```

**9. Write a program to calculate the m to the power n using recursion.**

```
def power(m, n):  
    if n == 0:  
        return 1  
    return m * power(m, n - 1)  
  
m = int(input("Enter base (m):"))  
n = int(input("Enter exponent (n):"))  
print(f"{power(m, n)}")
```

**10. Write a program to reverse a number using recursion.**

```
def reverse_number(n, rev=0):  
    if n == 0:  
        return rev  
    return reverse_number(n // 10, rev * 10 + n % 10)  
  
num = int(input("Enter a number:"))  
print(f"{reverse_number(num)}")
```