Assignment No.9

#WAP to find sum of following series using recursive functions:

#1!+2!+3!+4!+.....+n!   note:for fact and sum two recursive function

```
def fact(num):
    if(num == 0):
        return 1
    return num*fact(num - 1)
def sum_of_factorial(num):
    if(num == 1):
        return fact(1)
    return fact(num) + sum_of_factorial(num-1)
num=int(input("Enter number : "))
result=sum_of_factorial(num)
print("Factorial is :",result)
```

#Write a program to find factorial using recursion.

```
def fact(num):
    if(num == 0):
        return 1
    else:
        return num*fact(num - 1)
num=int(input("Enter number : "))
print("Factorial is :",fact(num))
```

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

```
def sum(num):
    if(num == 0):
        return 0
    else:
        return num + sum (num - 1)
```

```
num=int(input("Enter number:"))
print("Sum is :",sum(num))
```

#Write a program to find sum of digits using recursion.

```
def sum_of_digits(num):
    if(num == 0):
        return 0
    else:
        return num % 10 + sum_of_digits(num // 10)
num=int(input("Enter number of digit:"))
print("Sum of digit is:",sum_of_digits(num))
```

#Write a program to reverse a number using recursion.

```
def reverse(num,rev=0):
    if(num == 0):
        return rev
    else:
        return reverse(num // 10, rev * 10 + num % 10)
num=int(input("Enter number:"))
print("Reverse number:",reverse(num))
```

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

```
def prime_num(num):
    if num <= 1:
        return "Not a prime"
    for i in range(2, num // 2 + 1):
        if num % i == 0:
            return "Not a prime"
    return "Prime"
num = int(input("Enter number: "))
print(prime_num(num))
```

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

```
def count_digits(num):
    if(num == 0):
        return 0
   return 1 + count_digits(num // 10)
def armstrong_sum(num,power):
    if(num == 0):
        return 0
    return (num%10)**power+armstrong_sum(num//10,power)
def is_armstrong(num):
    power=count_digits(num)
    return num==armstrong_sum(num,power)
number = int(input("Enter a number: "))
if is_armstrong(number):
    print(f"{number} is an Armstrong number.")
else:
    print(f"{number} is not an Armstrong number.")
```

#Write a program to print Fibonacci series using recursion.

```
def fibonacci(n):
    if(n<=0):
        return 0
    elif(n==1):
        return 1
    else:
        return fibonacci(n-1) + fibonacci(n-2)
n = int(input("Enter number of terms: "))
fibonacci(n)
for i in range(n):
    print(fibonacci(i),end=' ')
```

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

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

#Write a program to reverse a given number using recursive function.

```
def reverse_number(n,rev=0):
    if n == 0:
        return rev
    else:
        return reverse_number(n // 10, rev * 10 + n % 10)
print(reverse_number(1234))
```