Assignment No:09

Q.1#write a program to find sum of following series using recursive function:

#i.1!+2!+3!+4!+.....+n!

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

def sum_of_factorail(n):
    if(n==1):
        return fact(1)
    return fact(n)+sum_of_factorail(n-1)

n=int(input("Enter a number:"))
if n<1:
    print("Enter number greater than 0.")
else:
    output=sum_of_factorail(n)
    print(f"The sum of the series 1!+2!+....+{n}!:{output} ")
```

Q.2.#wap 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//2)

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.")
```

Q.3.#wap to reverse a given number using recursive function

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

num=int(input("Enter the number:"))
reversed_num=reverse_number(num)
print(f'Reversed number:{reversed_num}')
```

Q.4.#wap to find sum of n numbers using recursion

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

num=int(input("Enter value of number:"))
if(num<0):
    print("Enter non-negative integer.")
else:
    total=sum(num)
    print(f'Sum of first {num} number is:{total}')
```

Q.5.#wap to find factorial using recursion.

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

```
num=int(input("Enter value of a number:"))
if(num<0):
    print("Factorial is not defined for negative number.")
else:
    output=factorial(num)
    print(f'The factorial of {num} is {output}')
```

Q.6.#wap 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)

t=int(input("Enter number of terms:"))
print("Fibonacci Series:")
for i in range(t):
    print(fibonacci(i),end=" ")
```

Q.7.#wap to find sum of digits using recursion

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

num=int(input("Enter a number:"))
output=sum_of_digits(num)
print("Sum of digits:",output)
```

Q.8.#wap 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))
```

Q.9.#wap 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 value of m:"))
n=int(input("Enter value of n:"))
output=power(m,n)
print(f'{m} to the power of {n} is {output}')
```

Q.10.#wap to reverse a number using recursion.

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

num=int(input("Enter the number:"))
reversed_num=reverse_number(num)
print(f'Reversed number:{reversed_num}')
```