Recursive Function:

==========

1.Write a recursive function to find the sum of series up to n integer number.

```
#include <stdio.h>

int sumSeries(int n){
   if(n==1)
      return 1;

   else
      return n+sumSeries(n-1);
}

int main(){

   int num,k;

   printf("Insert the number till to sum:");
   scanf("%d", &num);

   k=sumSeries(num);

   printf("The sum of the series :%d", k);

   return 0;
}
```

2.Write a recursive function to find the factorial of an integer number.

```
#include <stdio.h>

int factorial(int n){
   if(n==1)
      return 1;

   else
      return n*factorial(n-1);
}

int main(){

   int n,k;

   printf("Enter a number to find the factorial of:");
   scanf("%d", &n);

   k=factorial(n);

   printf("The factorial of the number %d is:%d",n,k);

   return 0;
}
```

3.Write a recursive function to count digit from user input integer number.

```
#include <stdio.h>

int countDigit(int n){
   if(n<=9)
      return 1;

   else
      return 1+countDigit(n/10);
}

int main(){
   int n,k;

   printf("Enter an integer number:");
   scanf("%d", &n);

   k=countDigit(n);
   printf("Total digits are:%d",k);

   return 0;
}
```

4.Write a recursive function to find the sum of all the digit from user input integer number.

_____

```
#include <stdio.h>

int addDigit(int n){
   if(n<=9)
      return n;
   else
   return (n%10+addDigit(n/10));
}

int main(){

   int n,k;

   printf("Enter an integer number:");
   scanf("%d", &n);

   k=addDigit(n);
   printf("Sum of all the digit:%d", k);

   return 0;
}
```

5.Write a recursive function for Fibonacci series.

```
#include <stdio.h>

int fibonacci(int n){

   if(n==0)
      return 0;

   else if(n==1)
      return 1;

   else
      return fibonacci(n-1)+fibonacci(n-2);
}

int main(){

   int num,k;

   printf("Enter n:");
   scanf("%d", &num);

   k=fibonacci(num);
   printf("Term %d of the Fibonacci series:%d", num,k);

   return 0;
}
```

6.Write a recursive function to reverse a integer number.

```
#include <stdio.h>
#include <math.h>

int reverse_number(int n){

   int mod;
   int digit=(int)log10(n);

   if(n==0)                          //or if(n<=9)
      return 0;              //      return n;

   else{
      mod=n%10;
      return (mod*pow(10,digit)+reverse_number(n/10));
   }
}
int main(){
   int k,num;
   printf("Enter a number to reverse:");
   scanf("%d", & num);

   k=reverse_number(num);

   printf("The reverse number is:%d", k);

   return 0;
}
```

7.Write some code to print all even numbers in a given range using recursion.

int EVEN (int start, int end)

```
#include <stdio.h>

int even(int start , int end){

   if(start>end)
      return 0;

   else if(start%2==0){
      printf("%d,", start);
      return even(start+2 , end);
   }
   else
      return even(start+1 , end);
}

int main(){

   int num_1,num_2,k;

   printf("Enter the starting number:");
   scanf("%d", &num_1);
   printf("Enter the ending number:");
   scanf("%d", &num_2);
```

```
   k=even(num_1,num_2);

   printf("All the even numbers between %d to %d are : %d",num_1 , num_2 , k);

return 0;

}
```

8.Write some code to find the sum of all natural numbers between 1 to n using recursion.

int SUM (int start, int end)

```
#include <stdio.h>

int sum(int start , int end){

   if(start>end)       //if(start==end)
      return 0;       //  return start; (it will be also ok)

   else
      return start+sum(start+1 , end);
}
```

```
int main(){

   int n1=1 , n2  ,k;

   printf("Enter the limit:");
   scanf("%d", &n2);

   k=sum(n1 , n2);

   printf("The sum of all natural numbers between %d to %d is :%d", n1,n2,k);

   return 0;
}
```

9.Write some code to check whether a number is palindrome or not using recursion.

int Palindrome (int num)

```
#include <stdio.h>
#include <math.h>

int reverseNumber(int num){

   int count_digit= (int)log10(num);

   if(num==0)
      return 0;
```

```
    else
        return (num%10 * pow(10,count_digit)+reverseNumber(num/10));
}

int palindrome(int num){

    if(num==reverseNumber(num))
        return 1;
}

int main(){

    int n;

    printf("Enter a number to check:");
    scanf("%d", & n);

    if(palindrome(n)==1)
        printf("\nThe number %d is a palindrome number",n);
    else
        printf("\nThe number %d is not a palindrome number",n);

    return 0;
}
```

10.Write some code to find GCD (HCF) of two numbers using recursion.

int GCD (int a, int b)

```
#include <stdio.h>

int GCD(int a, int b){
   if (b == 0)
      return a;
   else
      return GCD(b, a % b);
}

int main(){

   int num1,num2;

   printf("Enter the first number:");
   scanf("%d", &num1);
   printf("Enter the second number:");
   scanf("%d", &num2);

   printf("GCD of %d and %d is :%d ", num1, num2, GCD(num1, num2));

   return 0;
}
```

11.Write some code to find LCM of two numbers using recursion.

int LCM (int a, int b)

```
#include <stdio.h>

int lcm(int a, int b)
{
   static int multiple = 0;

   multiple += b;

   if((multiple % a == 0) && (multiple % b == 0))
   {
      return multiple;
   }
   else
   {
      return lcm(a, b);
   }
}

int main()
{
   int num1, num2, LCM;

   printf("Enter any two numbers to find lcm: ");
   scanf("%d%d", &num1, &num2);
```

```
    if(num1 > num2)
        LCM = lcm(num2, num1);
    else
        LCM = lcm(num1, num2);

    printf("LCM of %d and %d = %d", num1, num2, LCM);

    return 0;
}
```

12.Write some code to find maximum elements in an array using recursion.

int MAX(int arr[], int size)

```
#include <stdio.h>

int max_element(int arr[] , int size){

    static int i=0 , max=-9999;

    if(i<size){
        if(arr[i]>max){
```

```
        max=arr[i];
        i++;
        max_element(arr , size);
    }
      return max;
    }
}


int main(){

    int i,k,n;
    int num[100];

    printf("Enter how many numbers to store in the array:");
    scanf("%d", &n);

    for(i=0 ; i<n ; i++){

      printf("Enter num[%d]:", i);
      scanf("%d", &num[i]);
    }

    k= max_element(num , n);

    printf("The maximum element is: %d", k);

    return 0;
}
```

13.Write a recursive function that converts a decimal number to a binary number.

int DecToBin(int dec);

```
#include <stdio.h>

int DecToBin(int dec){

   if(dec==0)
      return 0;

   else
      return (dec%2)+(10*DecToBin(dec/2));
}

int main(){

   int k,num;

   printf("Enter a number for conversion:");
   scanf("%d", &num);

   k=DecToBin(num);

   printf("The number decimal number %d in binary: %d", num,k);

   return 0;
}
```

14.Write a recursive function that finds the sum of the following series.

1 + 1/2 + 1/4 + 1/8 + ... + (1/2)^n

```
#include <stdio.h>
#include <math.h>

float sumSeries(int n){

   if(n==0)
      return 1;

   else
      return 1/pow(2,n)+sumSeries(n-1);
}

int main(){

   int num;
   float k;

   printf("Enter the highest power:");
   scanf("%d", &num);

   k=sumSeries(num);

   printf("The sum of the series:%.2f", k);

   return 0;
}
```

15. Study the following sequence 3, 1, 3, 7, 11, 21, 39, 71, 131, 241, 443, 815, ...

In mathematical terms, the sequence Tn is defined as follows:

$$T_{n+3} = T_n + T_{n+1} + T_{n+2} \text{ for } n >= 0.$$

with initial values,

$$T_0 = 3, T_1 = 1, T_2 = 3,$$

Now take an integer input n from the user and print the sequence up to n. Use recursion for this problem.

```
#include <stdio.h>

int sumSeries(int n){

   if(n==0)
      return 3;
   else if(n==1)
      return 1;
   else if(n==2)
      return 3;
   else
      return sumSeries(n-1)+sumSeries(n-2)+sumSeries(n-3);
}

int main(){

   int num,k;

   printf("Enter the highest term:");
```

```
    scanf("%d", &num);

    k=sumSeries(num);

    printf("The sum of the series:%d", k);

    return 0;
}
```

16. Study the following sequence

    1/1! +2/2! +3/3! + ... +n/n!

Now, use recursion to print the summation of the given series up to nth term.

```
#include <stdio.h>

int factorial(int n){

    if(n==1)
        return 1;

    else
        return n*factorial(n-1);
}

float sumSeries(int term){
```

```
    if(term==1)
        return 1;

    else
        return (term/factorial(term))+sumSeries(term-1);
}

int main(){

    int num;
    float k;

    printf("Enter the highest term:");
    scanf("%d", &num);

    k=sumSeries(num);

    printf("The sum of the series:%.2f", k);

    return 0;
}
```

17. What will be the following recursion function output for Fn[20]

```
Int Fn(int n){
        If(n<=2)
                Retutn 2;
        Else
                Return n+Fn(n-1);
        }
```

Modify this function to add every odd numbers up to n.(Sir final)

---

output:

_____

2+3+4+5+6+7+.....+19+20=209

```
#include <stdio.h>

int Fn(int n){

  if(n==1)
    return 1;

  else if(n%2!=0){
    return n+Fn(n-2);
  }
```

```
    else{
        return Fn(n-1);
    }
}

int main(){

    int num,k;

    printf("Enter the ending number:");
    scanf("%d", &num);

    k=Fn(num);

    printf("The sum of all numbers up to %d is :%d",num, k);

    return 0;
}
```

18.Write a program in C to check a number is a prime number or not using recursion.

```
#include <stdio.h>

int prime(int num){

   static int i=2;

   if(i<=num/2){
      if(num%i==0){
         return 1;
      }
      else{
         i++;
         return prime(num);
      }
   }
}

int main(){

   int num;

   printf("Enter a number to check:");
   scanf("%d", &num);
```

```
   if(prime(num)==1)
      printf("The number %d is a not prime number",num);
   else
      printf("The number %d is a prime number",num);

   return 0;
}
```

19. Write a recursive function that converts a decimal number to a octal number.

int DecToOct(int dec);

```
#include <stdio.h>

int decToOct(int dec){

   if(dec==0)
      return 0;

   else
      return (dec%8)+10*decToOct(dec/8);
}
```

```
int main(){

   int decimal,k;

   printf("Enter a decimal number:");
   scanf("%d", &decimal);

   k=decToOct(decimal);

   printf("The number in octal:%d",k);

   return 0;
}
```