Assignment - 13

1. Write a recursive function to calculate sum of first N natural numbers

```
#include<stdio.h>

int addNumbers(int);

int main()
{

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum = %d", addNumbers(num));
  return 0;
}

int addNumbers(int n)
{
  if (n != 0)
    return n + addNumbers(n - 1);
  else
    return n;
}
```

2. Write a recursive function to calculate sum of first N odd natural numbers

```
#include<stdio.h>

int sumOddNumbers(int);
```

```
int main()
{

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum of Odd natural numbers = %d", sumOddNumbers(num));
  return 0;
}

int sumOddNumbers(int n)
{
  if (n == 1)
    return 1;
  else
    return (2*n-1 + sumOddNumbers(n - 1));
}
```

3. Write a recursive function to calculate sum of first N odd natural numbers

```
#include<stdio.h>

int sumEvenNumbers(int);

int main()
{

  int num;
```

```
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum of Even natural numbers = %d",
sumEvenNumbers(num));
  return 0;
}

int sumEvenNumbers(int n)
{
  if (n == 1)
    return 2;
  else
    return (2*n) + sumEvenNumbers(n - 1);
}
```

4. Write a recursive function to calculate sum of squares of first n natural numbers

```
#include<stdio.h>

int sumOfSquareNumbers(int);

int main()
{

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum of Square natural numbers = %d",
sumOfSquareNumbers(num));
```

```
  return 0;
}

int sumOfSquareNumbers(int n)
{
  if (n == 1)
    return 1;
  else
    return (n*n) + sumOfSquareNumbers(n - 1);
}
```

5. Write a recursive function to calculate sum of digits of a given number

```
#include<stdio.h>

int sumOfDigitNumbers(int);

int main()
{

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Sum of Digit numbers = %d", sumOfDigitNumbers(num));
  return 0;
}
```

```
int sumOfDigitNumbers(int n)
{
  if (n/10 == 0)
    return n;
  else
    return (n%10) + sumOfDigitNumbers(n/10);
}
```

6. Write a recursive function to calculate factorial of a given number

```
#include<stdio.h>

int Factorial(int);

int main()
{

  int num;
  printf("Enter a positive integer: ");
  scanf("%d", &num);
  printf("Factorial Number = %d", Factorial(num));
  return 0;
}

int Factorial(int n)
{
  if (n > 0)
    return n * Factorial(n - 1);
  else
    return 1;
}
```

7. Write a recursive function to calculate HCF of two numbers

```
#include<stdio.h>

int HCF(int ,int );
int main()
{
    int x,y,result;
    printf("Enter two number x and y : ");
    scanf("%d %d",&x,&y);
    result = HCF(x,y);
    printf("HCF = %d",result);
    return 0;
}
int HCF(int a,int b)
{
    if(a==b)
        return a;
    if(a%b==0)
        return b;
    if(b%a==0)
        return a;
    if(a>b)
        return (HCF((a%b),b));
    else
        return (HCF(a,(b%a)));
}
```

8. Write a recursive function to print first N terms of Fibonacci series

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

```
int fibbonacci(int);
int main()
{
    int i,n;
    printf("Enter term of fibbonacci : ");
    scanf("%d",&n);
    printf("Fibbonacci series : ");
    for(i=1;i<=n;i++)
        printf("%d ",fibbonacci(i));
    return 0;
}
int fibbonacci(int n)
{
    if(n==1||n==2)
        return 1;
    return (fibbonacci(n-1)+fibbonacci(n-2));
}
```

9. Write a program in C to count the digits of a given number using recursion

```
                #include <stdio.h>
int countDigits(int num);

int main()
{
    int number;
    int count=0;

    printf("Enter a positive integer number: ");
    scanf("%d",&number);
```

```
    count=countDigits(number);

    printf("Total digits in number %d is: %d\n",number,count);

    return 0;
}

int countDigits(int num)
{
    static int count=0;

    if(num>0)
    {
        count++;
        countDigits(num/10);
    }
    else
    {
        return count;
    }
}
```

10. Write a program in C to calculate the power of any number using recursion.

```
                #include <stdio.h>
int power(int n1, int n2);
int main()
{
    int base, a, result;
    printf("Enter base number: ");
    scanf("%d", &base);
```

```
    printf("Enter power number(positive integer): ");
    scanf("%d", &a);
    result = power(base, a);
    printf("%d^%d = %d", base, a, result);
    return 0;
}

int power(int base, int a) {
    if (a != 0)
        return (base * power(base, a - 1));
    else
        return 1;
}
```