More on Recursion in C Language

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

```c
#include <stdio.h>
int sumNatural(int);
int main()
{
    int n, add;
    printf("Enter a number : ");
    scanf("%d", &n);
    add = sumNatural(n);
    printf("Sum of %d natural numbers is %d", n, add);
    return 0;
}

// below function is to calculate sum of n natural numbers and return
int sumNatural(int x)
{
    if (x == 1)
        return 1;
    return (x + sumNatural(x - 1));
}
```

```
==========================================================================
Output:
Enter a number : 8
Sum of 8 natural numbers is 36
```

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

```c
#include <stdio.h>
int oddNatural(int);
int main()
{
    int n, add;
    printf("Enter a number : ");
    scanf("%d", &n);
    add = oddNatural(n);
    printf("Sum of %d odd natural numbers is %d", n, add);
    return 0;
}

// below function is to calculate sum of n odd natural numbers and return
int oddNatural(int x)
{
    if (x == 1)
        return 1;
    return ((x * 2 - 1) + oddNatural(x - 1));
}
```

```
==========================================================================
Output:
Enter a number : 3
Sum of 3 odd natural numbers is 9
```

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

```c
#include <stdio.h>
int evenNatural(int);
int main()
{
    int n, add;
    printf("Enter a number : ");
    scanf("%d", &n);
    add = evenNatural(n);
```

```
    printf("Sum of %d even natural numbers is %d", n, add);
    return 0;
}

// below function is to calculate sum of n even natural numbers and return
int evenNatural(int x)
{
    if (x == 1)
        return 2;
    return ((x * 2) + evenNatural(x - 1));
}
```

```
Output:
Enter a number : 5
Sum of 5 even natural numbers is 30
```

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

```
#include <stdio.h>
int sumNaturalsquare(int);
int main()
{
    int n, add;
    printf("Enter a number : ");
    scanf("%d", &n);
    add = sumNaturalsquare(n);
    printf("Sum of square of %d natural numbers is %d", n, add);
    return 0;
}

// below function is to calculate sum of square of n natural numbers and
return
int sumNaturalsquare(int x)
{
    if (x == 1)
        return 1;
    return (x * x + sumNaturalsquare(x - 1));
}
```

```
Output:
Enter a number : 7
Sum of square of 7 natural numbers is 140
```

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

```
#include<stdio.h>
int sumDigit(int);
int main()
{
    int n, sum;
    printf("Enter a number : ");
    scanf("%d", &n);
    sum = sumDigit(n);
    printf("Sum of digits is %d", sum);
    return 0;
}
// below recursive function is calculate sum of digits and return value
int sumDigit(int x)
{
    int rem;
    rem = x % 10;
    if(x == 0)
        return 0;
    return(rem + sumDigit(x / 10));
}
```

```
================================================================================
Output:
Enter a number : 597
Sum of digits is 21
```

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

```c
#include<stdio.h>
int fact(int);
int main()
{
    int n, result;
    printf("Enter a number : ");
    scanf("%d", &n);
    result = fact(n);
    printf("Factorial of %d\n", n);
    printf("%d! = %d", n, result);
    return 0;
}
// below recursive function is to calculate factorial of a given number
int fact(int x)
{
    if(x == 0)
        return 1;
    return(x * fact(x - 1));
}
```

```
================================================================================
Output:
Enter a number : 10
Factorial of 10
10! = 3628800
```

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

```c
#include <stdio.h>
int HCF(int, int);
int main()
{
    int a, b, result;
    printf("Enter two numbers : ");
    scanf("%d%d", &a, &b);
    result = HCF(a, b);
    printf("HCF of %d and %d is %d", a, b, result);
    return 0;
}
// below recursive function is to calculate HCF of two numbers by ecluis
theorem
int HCF(int x, int y)
{
    if (x == y)
        return x;
    if (x % y == 0)
        return y;
    if (y % x == 0)
        return x;
    if (x > y)
        return (HCF(x % y, y));
    else
        return (HCF(x, y % x));
}
```

```
================================================================================
Output:
Enter two numbers : 36 42
HCF of 36 and 42 is 6
```

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

```c
#include <stdio.h>
void printFibonacci(int);
int main()
{
    int n;
    printf("Enter a number : ");
    scanf("%d", &n);
    printf("Fibonacci Series\n");
    printFibonacci(n);
    return 0;
}
// below recursive function is to print first N terms of Fibonacci series
void printFibonacci(int x)
{
    static int a = -1, b = 1, s;
    if (x > 0)
    {
        s = a + b;
        a = b;
        b = s;
        printf("%d ", s);
        printFibonacci(x - 1);
    }
}
```

```
Output:
Enter a number : 6
Fibonacci Series
0 1 1 2 3 5
```

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

```c
#include <stdio.h>
int countDigit(int);
int main()
{
    int n;
    printf("Enter a number : ");
    scanf("%d", &n);
    printf("%d digit number", countDigit(n));
    return 0;
}
// below recursive function is to count the digits of a given number using
recursion.
int countDigit(int x)
{
    if (x < 10)
        return 1;
    else
        return (1 + countDigit(x / 10));
}
```

```
Output:
Enter a number : 23578
5 digit number
```

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

```c
#include <stdio.h>
int power(int, int);
int main() {
    int base, a, result;
    printf("Enter any 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;
}
// below recursive function is to calculate the power of any number.
int power(int base, int a) {
    if (a != 0)
        return (base * power(base, a - 1));
    else
        return 1;
}
```

==========

```
Output:
Enter any number: 3
Enter power number(positive integer): 4
3^4 = 81
```