Assignment - 13

More on Recursion in C Language

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

```
#include<stdio.h>
int sum(int);
int main()
{
   int n;
   printf("\nEnter N :");
   scanf("%d",&n);
   printf("\n%d",sum(n));
}
int sum(int n)
{
   if (n != 0)
      return n + sum(n - 1);
   else
      return n;
}
```

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

```
#include<stdio.h>
int sum(int);
int main()
{
   int n;
   printf("\nEnter N :");
   scanf("%d",&n);
   printf("\n%d",sum(n));
}
int sum(int n)
{
            if (n==1)
            {
               return 1;
            }
            else
```

```
        {
            return 2*n-1 + sum(n-1);
        }
}
```

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

```
#include<stdio.h>
int sum(int);
int main()
{
   int n;
   printf("\nEnter N :");
   scanf("%d",&n);
   printf("\n%d",sum(n));
}
int sum(int n)
{
        if (n==1)
        {
            return 1;
        }
        else
        {
            return 2*n-1 + sum(n-1);
        }
}
```

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

```
#include<stdio.h>
int sum(int);
int main()
{
   int n;
   printf("\nEnter number: ");
   scanf("%d", &n);
   printf("Sum of sqrs of first %d natural numbers is %d",n,sum(n));
   return 0;
}
int sum(int n)
```

```
{
   if (n==0)
   {
      return 0;
   }
   else
   {
      return (n)*(n) + sum(n-1);
   }
}
```

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

```
#include<stdio.h>
int sum(int);
int main()
{
   int n;
   printf("\nEnter number : ");
   scanf("%d",&n);
   printf("sum of digits of a %d is %d",n,sum(n));
}
int sum(int n)
{
   if (n==0)
   {
      return 0;
   }
   else
   {
      return n%10 + sum(n/10);
   }
}
```

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

```
#include<stdio.h>
int fact(int);
int main()
{
   int n = 5;
```

```
   printf("\nfactorial of a %d is %d",n,fact(n));
   return 0;
}
int fact(int n)
{
   if (n==1)
   {
      return 1;
   }
   else
   {
      return n*fact(n-1);
   }
}
```

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

```
#include<stdio.h>
int hcf(int,int);
int main()
{
   int n1 = 18, n2 = 27;
   printf("\n%d",hcf(n1,n2));
}
int hcf(int n1, int n2)
{
   int c;
   int num1 = n1<n2 ? n1:n2;
   int num2 = n1>n2 ? n1:n2;
   if (num2%num1==0)
   {
      return num1;
   }
   else
   {
      int num3 = num2%num1;
      int num4 = num1;
      c = hcf(num3, num4);
      return c;
   }
}
```

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

```
#include<stdio.h>
int fibo(int);
int main()
{
   int n;
   printf("\nEnter number: ");
   scanf("%d",&n);
   fibo(n);
}
int fibo(int n)
{
   static int n1=0,n2=1,n3;
   if(n>0)
   {
       n3 = n1 + n2;
       n1 = n2;
       n2 = n3;
       printf("%d ",n3);
       fibo(n-1);
   }
}
```

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

```
#include<stdio.h>
int digits(int);
int main()
{
   int n = 123456;
   printf("\ndigits of a given number is %d",digits(n));
}
int digits(int n)
{
   int count=0;
   if (n!=0)
   {
      count++;
      return count + digits(n/10);
```

```
    }
    else
        return 0;
}
```

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

```
#include<stdio.h>
int pow(int n1, int n2);
int main()
{
    int base, a;
    printf("\nEnter base and its power number: ");
    scanf("%d %d",&base,&a);
    printf("the power of %d^%d is %d",base,a,pow(base,a));
}
int pow(int base, int a)
{
    if (a!=0)
    {
        return base * pow(base,a-1);
    }
    else
    {
        return 1;
    }
}
```