Write a program in C

1. To print first 50 natural numbers using recursion.

```
#include <stdio.h>

void print_numbers(int n)
{
   if (n <= 50)
   {
      printf("%d ", n);
      print_numbers(n+1);
   }
}

int main()
{
   print_numbers(1);
   return 0;
}
```

2. To calculate the sum of Natural Numbers Using Recursion.

```
#include <stdio.h>

int sum_natural(int n)
{
   if (n == 1)
      return 1;
   else
      return n + sum_natural(n-1);
}

int main()
{
   int n;
   printf("Enter a positive integer: ");
   scanf("%d", &n);
```

```
    printf("Sum of first %d natural numbers: %d", n,
sum_natural(n));
    return 0;
}
```

3. To generates the Fibonacci series for a given number using a recursive function.

```
#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 n, i;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Fibonacci series up to %d:\n", n);
    for (i = 0; i < n; i++)
        printf("%d ", fibonacci(i));
    return 0;
}
```

4. To count the digits of a given number using recursion.

```
#include <stdio.h>

int count_digits(int n)
{
    if (n == 0)
        return 0;
    else
        return 1 + count_digits(n/10);
}

int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Number of digits in %d: %d", n, count_digits(n));
    return 0;
}
```

5. To find the sum of digits of a number using recursion.

```
#include <stdio.h>

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

int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Sum of digits in %d: %d", n, sum_digits(n));
    return 0;
}
```

6. To find GCD of two numbers using recursion.

```
#include <stdio.h>

int gcd(int a, int b)
```

```
{
    if (b == 0)
        return a;
    else
        return gcd(b, a % b);
}

int main()
{
    int a, b;
    printf("Enter two positive integers: ");
    scanf("%d %d", &a, &b);
    printf("GCD of %d and %d: %d", a, b, gcd(a, b));
    return 0;
}
```

7. To convert a decimal number to binary using recursion.

```
#include <stdio.h>

void dec_to_bin(int n)
{
    if (n == 0)
        return;
    dec_to_bin(n/2);
    printf("%d", n%2);
}

int main()
{
    int n;
    printf("Enter a positive integer: ");
    scanf("%d", &n);
    printf("Binary representation of
```

8. To calculate the power of any number using recursion.

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

```
int power(int x, int y)
{
    if(y == 0)
        return 1;
    else if(y%2 == 0)
        return power(x*x, y/2);
    else
        return x * power(x*x, y/2);
}

int main()
{
    int x,y;
    printf("Enter the base: ");
    scanf("%d",&x);
    printf("Enter the exponent: ");
    scanf("%d",&y);
    printf("%d^%d = %d",x,y,power(x,y));
    return 0;
}
```

9. To calculate the factorial of natural numbers till a range inputted by user by recursion.

```
#include<stdio.h>

int factorial(int n)
{
    if(n == 0)
        return 1;
    else
        return n * factorial(n-1);
}

int main()
{
    int n,i;
    printf("Enter the range of numbers: ");
    scanf("%d",&n);
    printf("Factorials:\n");
    for(i=1;i<=n;i++)
        printf("%d! = %d\n",i,factorial(i));
    return 0;
}
```