Q1.Write a recursive function to calculate sum of first N natural numbers

```c
#include <stdio.h>

int rec(int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    printf("sum is = %d",rec(num));

    return 0;
}

int rec(int num)
{
    if (num == 1)
        return num;
    else
        return (num + rec(num - 1));
}
```

Q2.Write a recursive function to calculate sum of first N odd natural numbers

```c
#include <stdio.h>

int rec(int num, int odd);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    printf("Sum is = %d", rec(num, 1));
    return 0;
}

int rec(int num, int odd)
{
    if (num == 1)
        return odd;
    else
    {
        odd = odd + rec(num-1, odd + 2);
        return odd;
    }
}
```

Q3.Write a recursive function to calculate sum of first N even natural numbers

```c
#include <stdio.h>

int rec(int, int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    printf("Sum is = %d",rec(num, 2));

    return 0;
}

int rec(int num, int even)
{
    if (num == 1)
        return even;
    else
    {
        even = even + rec(num - 1, even + 2);
        return even;
    }
}
```

Q4.Write a recursive function to calculate sum of squares of first n natural numbers

```c
#include <stdio.h>

int rec(int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    printf("Sum  is = %d", rec(num));

    return 0;
}

int rec(int num)
{
    if (num == 1)
        return 1;
    else
        return ((num * num) + rec(num - 1));
}
```

Q5.Write a recursive function to calculate sum of digits of a given number

```c
#include <stdio.h>

int rec(int);

int main(int argc, char *argv[])
{
    printf("sum is = %d",rec(5));

    return 0;
}

int rec(int num)
{
    if (num == 1)
        return 1;
    else
        return (num + rec(num - 1));
}
```

Q6.Write a recursive function to calculate factorial of a given number

```c
#include <stdio.h>

int rec(int);

int main(int argc, char *argv[])
{
    printf("factorial = %d",rec(5));

    return 0;
}

int rec(int num)
{
    if (num == 2)
        return 2;
    else
        return (num * rec(num - 1));
}
```

Q7.Write a recursive function to calculate HCF of two numbers

```c
#include <stdio.h>

void rec(int a, int b, int div, int ans);

int main(int argc, char *argv[])
{
    int a, b;

    printf("Enter to number = ");
    scanf("%d %d", &a, &b);

    rec(a, b, 2, 1);

    return 0;
}

void rec(int a, int b, int div, int ans)
{
    if (div > a)
    {
        printf("HCF = %d", ans);
    }
    else
    {
        if ((a % div == 0) && (b % div == 0))
        {
            ans = ans * div;
            rec(a / div, b / div, div, ans);
        }
        else
        {
            div = div + 1;
            rec(a, b, div, ans);
        }
```

```
        }
    }
```

Q8.Write a recursive function to print first N terms of Fibonacci series

```c
#include <stdio.h>

void rec(int, int, int, int);

int main(int argc, char *argv[])
{
    int num;

    printf("Enter number = ");
    scanf("%d", &num);

    printf("1 ");
    rec(num, 0, 1, 0);

    return 0;
}

void rec(int num, int a, int b, int fact)
{
    if (num == 1)
    {
        return;
    }
    else
    {
        num = num - 1;

        fact = a + b;

        printf("%d ", fact);

        a = b;
        b = fact;
```

```
            rec(num, a, b, fact);
        }
    }
```

Q9.Write a program in C to count the digits of a given number using recursion.

```c
#include <stdio.h>

void rec(int, int);

int main(int argc, char *argv[])
{
    rec(12345,0);

    return 0;
}

void rec(int num, int count)
{
    if (num == 0)
    {
        printf("total number = %d", count);
    }
    else
    {
        count = count + 1;
        rec(num/10, count);
    }
}
```

Q10.Write a program in C to calculate the power of any number using recursion.

```c
#include <stdio.h>

int rec(int, int);

int main(int argc, char *argv[])
{
    int num, pow;

    printf("Enter number = ");
    scanf("%d", &num);

    printf("Enter power = ");
    scanf("%d", &pow);

    printf("Answer = %d",rec(num, pow));

    return 0;
}

int rec(int num, int pow)
{
    if (pow == 1)
        return num;
    else
        return (num * rec(num, pow-1));
}
```